Direct Numerical Simulations of Unsteady Separated Flows Using Vortex Methods

Thesis by

Petros D. Koumoutsakos

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California 91125

1993

(Defended 6th August 1992)

ITHAKA

As you start your journey for Ithaka
wish your trip is long,
full of adventure and discovery.
The Laestrygonas and the Cyclopas,
the wrathful Poseidona - don't let them scare you:
they'll never block your way,
as long as your thoughts are free,
as long as rare excitement stirs your senses and your body.
The Laestrygonas and the Cyclopas,
the wrathful Poseidona - you 'll never face them
unless they ride along inside you,
unless your soul raises them before you.

Wish your journey is long.
Many the summer mornings when,
with what pleasure, what joy,
you enter harbors you're seeing for the first time;
may you stop at Phoeniciens seaports
and get the finest wares,
mother of pearl and coral, much amber and ebony
and sensual scents of every kind,
as many sensual perfumes as you can;
and may you travels reach many Egyptian cities,
to learn and keep on learning from their scholars.

Keep Ithaka always on your mind.
Arriving there is what you're destined for.
But don't hurry the journey at all.
Better if it lasts for years,
so that you are old when you cast your anchor in the island,
rich with all you've gained on the way,
not expecting any wealth from Ithaka.

Ithaka gave you the fine journey.
Without Ithaka you wouldn't have taken the road.
But she has nothing more to give you now.

And if you find her poor, Ithaka did not deceive you.
Wise as you now are , with such experience,
you have already understood what an Ithaka really is.

A. Kavafis (1911).

Acknoweledgments

I would like to express my deepest gratitude to my advisor, Professor Anthony Leonard, who with his unique dedication and kindness saw me through every aspect of this work. I would also like to thank him in particular for the freedom he allowed in the course of this investigation and the ways he showed me to approach scientific as well as real life phenomena. It is a pleasure and honor being his student.

I wish to thank all the members of the Caltech community and in particular the people of GALCIT for making the Caltech adventure a memorable one. For lending their time, interest and insight I wish to sincerely thank the members of my Examining Committee: Pr. Hans Hornung, Pr. James Knowles, Pr. Dale Pullin and Pr. Anatol Roshko. My sincere thanks also to Pr. Paul Dimotakis for his availability and the many inspirational discussions; to Jean Anderson and Pat Gladson for their friendly and irreplaceable assistance in the library; to the people involved with the management of the computer systems in GALCIT and in particular to Dan Lang for being the good wizard of computers; to Riccardo Bonazza for sharing his computer graphics expertise; and to Sylvie Mas-Gallic for many enlightening conversations.

Among the several contributions made to this thesis and the author himself, both personal and scientific, I wish to distinguish those made by François Pépin. The present work has benefited tremendously by his pathbreaking investigations and I would always remain indebted as he has been a kind mentor and an invaluable friend throughout these years.

I want to thank my friends, whose love and warmth is what kept me going many times these years. Yiannis Anastasiadis for his sense of humour and for sharing so many trips up and down freeways (alas, except one) and most of all for being the "mouloxtos" that he is. Christos Balaskas for always caring and always being a friend no matter the time and the distance. Daphne Bouzikou, for the late nights when she was trying to chase the bugs out of my programs and for standing by me and really sharing the roughest part of the road. Nikos Papadakos for all those twenty years (and their respective summers), and for always checking that I keep my promises.

Finally, its my deepest pride to thank here my mother and father. My journey has been their journey and this thesis is dedicated to their love.

Abstract

Numerical simulations are presented for viscous incompressible flows with and without solid wall boundaries. Our numerical method is based on vortex methods. The classical inviscid scheme is enhanced to account for viscous effects via the method of particle strength exchange. The method is extended to account for the enforcement of the no-slip boundary condition as well by appropriately modifying the strength of the particles. Computations are possible for extended times by periodically remeshing the vorticity field.

The particles are advanced using the Biot-Savart law for the evaluation of the velocity. Computations are made using up to $\mathcal{O}(10^6)$ vortex particles by efficiently implementing the method of multipole expansions for vector computer architectures to obtain an $\mathcal{O}(N)$ algorithm.

The method is used to simulate the inviscid evolution of an elliptical vortex in an unbounded fluid as well as unsteady separated flows around circular cylinders for a wide range of Reynolds numbers (40 - 9500). Direct comparisons are made of the results of the present method with those from a variety of theoretical, computational and experimental studies. The results exhibit the robustness and validity of the present method and allow to gain physical insight as to vorticity formation and its relation to the forces experienced by the body

Table of Contents

CHAPTER 1

Introduction

This thesis is concerned with the development of a Lagrangian numerical scheme, based on vortex methods, for the direct numerical simulation of unsteady, incompressible, viscous flows. The dynamics of such flows are determined by the evolution of their vorticity field. In vortex methods all of the vorticity is attached to the computational elements (vortex particles) that are convected with the local fluid velocity. Unlike grid methods no computational elements are devoted to the irrotational part of the flow. This is an attractive feature of the method as in many engineering flows of interest (such as external aerodynamics) vorticity occupies only a small fraction of the domain.

The straightforward physical interpretation and the adaptive character of vortex methods have inspired many studies since the pioneering work of Rosenhead in 1931. The Lagrangian character of the method has made it a viable alternative to finite difference and spectral methods for simulations of flows around complex configurations. However vortex methods have primarily been used only as an engineering tool in the past to gain physical insight and a rough estimate of unsteady forces for a large class of separated flows. A thorough review of the foundations of method and an extensive account of previous applications of the method may be found in review articles by Leonard (1980,1985) and Sarpkaya (1989).

Vortex methods have not had much success in the past in competing with grid based schemes in the arena of direct numerical simulation. The adaptive properties of vortex methods have been counterbalanced by their cost and their difficulty in representing accurately viscous effects. This thesis focuses on these limitations of vortex methods. A rigorous method is developed that alleviates these restrictions while maintaining the attractive features of the method.

The velocities of the fluid elements can be computed using Biot - Savart's law. For N computational elements this requires $\mathcal{O}(N^2)$ operations per time step using the classical algorithm. This has limited in the past the number of computational elements that one may use to a few thousand even with the recent advent of modern supercomputers. To circumvent this problem hybrid methods (such as CIC (Christiansen, 1973)) have been used to compute the velocity field. In these schemes a grid is superposed on the particles and a streamfunction is computed by solving a Poisson's equation (with vorticity as the source term) on the grid. The velocity field is computed (as a curl of the streamfunction) on the grid as well and is subsequently interpolated onto the particles. The cost is proportional then to $\mathcal{O}(N + M\log M)$, where M is the number of mesh points on the grid. However this scheme carries along with its efficiency all the drawbacks of grid methods that vortex methods were designed to circumvent. Otherwise absent, numerical diffusion is introduced in the computation of the velocity field, the far field condition (built in the Biot - Savart law) needs to be taken specifically into account and finally the applicability of the method is reduced to flows around regular geometries. Recent advances have renewed the interest in using the Biot Savart law for the computation of the velocities. Fast summation algorithms have been developed, based on multipole expansions (Greengard and Rohklin, 1987) that allow computations at an $\mathcal{O}(N)$ cost per time step. The present efficient implementation of these schemes on vector computer architectures allows for simulations using $\mathcal{O}(10^6)$ particles at a reasonable computational cost.

The Lagrangian nature of the method encounters difficulties in dealing with viscous effects. The method is faced with the challenge of approximating the smooth diffusion operator with information gathered from the (possibly) scattered particle locations. Twenty years ago Chorin (1973) proposed to add a random walk to the motion of the particles to account for diffusion. This scheme has found extensive use in the past but has the disadvantage of a low convergence rate (proportional to $1/\sqrt{N}$). More recently an alternative scheme has been introduced (Raviart, 1987) that accounts for viscosity by appropriately modifying the strength of the particles. This method of particle strength exchange (PSE) has been shown to have good convergence properties but its accuracy is heavily affected by the distortion of the computational mesh. In the present work the scheme is complemented with a particle redistribution algorithm that alleviates this problem and allows for accurate viscous simulations for extended times.

The grid-free character of the method makes it convenient to simulate flows

around arbitrarily complex configurations. However the use of a vorticity formulation complicates the enforcement of the (natural) velocity boundary conditions (the fluid must adhere to the wall) for the governing Navier Stokes equations. In Chorin's algorithm new vortex elements are inserted at the boundary and subsequently undergo a random walk to mimic the vorticity generation due to the no-slip walls. However for this technique there is a large number of parameters that one needs to adjust to obtain meaningful results and the scheme bears the low convergence rate of the random walk for diffusion. In this thesis a novel technique for the enforcement of the no-slip boundary condition is presented. The no-slip condition is shown to be equivalent to a vorticity flux at the wall. This vorticity flux is subsequently distributed to the particles. No new particles are created but the strengths of the existing ones are modified so that the velocity boundary condition is enforced. This scheme is complementary to the scheme of the PSE and presents a natural algorithm for the creation of vorticity in the fluid due to solid walls.

The result is a computational scheme that is capable of producing accurate direct numerical simulations for bounded and unbounded flows. The scheme is applied to the flow resulting from an impulsively started rotating and translating circular cylinder. This problem is studied as a prototype for unsteady separated flows and has been the subject of several experimental and theoretical works. Experimental studies of such flows date back to the works of Prandtl (1925). The problem of the impulsively started cylinder was first studied by Blasius in 1908 and computations of such flows started appearing in the late 50's (Payne, 1958). The flow has been investigated extensively in experiments of Bouard and Coutanceau (1980) and Coutanceau and Ménard (1985). Many computational works have appeared on this subject in the last ten years as well (Badr and Dennis (1985), Chen, Ou and Pearlstein (1992) etc.) helping us to obtain information unavailable by experiments. This work adds to these simulations as the present scheme provides an accurate description of the vorticity field and helps us gain some insight as to the vorticity production at solid surfaces and its relation to such quantities such as the drag and lift coefficient of the body.

The thesis is organized as follows:

Chapter 2 of the thesis presents the governing equations and boundary conditions and their transformation to a form tractable by the present method. The numerical method is outlined and the basic features of the algorithm are presented. *Chapter*

3 describes the fast multipole method for the computation of the velocity field and its efficient implementation on vector computers. *Chapter 4* addresses the issue of the treatment of viscous effects by vortex methods. The scheme of particle strength exchange is presented and is compared to several numerical schemes for the solution of the diffusion equation to asses its validity and efficiency. *Chapter 5* is concerned with the enforcement of velocity boundary conditions in the context of vortex methods. A novel algorithm of vorticity creation is presented that rigorously enforces the no - slip condition. *Chapter 6* complements the previous two chapters. It tackles the problem of particle redistribution so that viscous computations are possible for extended times. *Chapter 7* discusses the application of the numerical scheme for the simulation of vortical flows. We examine the inviscid evolution of a specific elliptical vorticity distribution and subsequently focus on the simulations of the early developments in the wake of an impulsively started/stopped translating and rotating cylinder. These simulations serve the twofold purpose of validating our method and help us gain insight as to the generation of vorticity on the surface of the body and its relation to such quantities as the drag and lift coefficient on the body. Extensive comparisons are made with previous experimental, theoretical and computational works. Finally in *Chapter 8* we overview the present work and its results and conclude by suggesting possible extensions and applications of the method. We also address some issues that need to be considered in future investigations.

CHAPTER 2

Navier-Stokes Equations and Vortex Methods

2.1 The 2-D Navier-Stokes Equations

Two-dimensional incompressible flow of a viscous fluid may be described in terms of the velocity ($\mathbf{u}(\mathbf{x},t)$) and the pressure ($p(\mathbf{x},t)$) of the flow by the momentum equation :

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} \tag{2.1a}$$

and the continuity equation :

$$\nabla\cdot\mathbf{u} = 0. \tag{2.1b}$$

Here ν,ρ denote the kinematic viscosity and the density of the fluid. To define a certain flow the equations should be supplemented with initial conditions :

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \tag{2.1c}$$

When the flow is around a solid configuration (translating with velocity $\mathbf{U_b}(t)$ and rotating around its center of mass ($\mathbf{x}_b$) with angular velocity $\Omega(t)$), boundary conditions need to be enforced. On the surface of the body ($\mathbf{x}_s$) the velocity of the fluid ($\mathbf{u}$) should be equal to the velocity of the body ($\mathbf{U_s}$):

$$\mathbf{u}(\mathbf{x}_s) = \mathbf{U_s} \tag{2.1d}$$

with :

$$\mathbf{U_s} = \mathbf{U_b}(t) + \Omega(t)\,\hat{\mathbf{e}_z}\times(\mathbf{x}_s - \mathbf{x}_b)$$

and at infinity :

$$\mathbf{u}(\mathbf{x}) \to \mathbf{U}_\infty \qquad \text{as} \quad |\mathbf{x}| \to \infty\,,$$

where $\mathbf{U}_\infty$ is the free stream velocity. For the purposes of the present numerical scheme equation Eq. 2.1a is recast in terms of the vorticity ($\omega = \omega\hat{\mathbf{e}_z}$):

$$\omega = \nabla\times\mathbf{u}\,. \tag{2.2}$$

Taking the curl of Eq. 2.1a and using the definition of the vorticity results in :

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \nabla^2 \omega \tag{2.3}$$

with initial conditions

$$\omega(\mathbf{x}, 0) = \nabla \times \mathbf{u}_0(\mathbf{x}) .$$

Furthermore we associate a streamfunction $\boldsymbol{\Psi}(\mathbf{x}) = \Psi \widehat{\mathbf{e}_z}$ with the velocity field so that :

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi} \tag{2.4}$$

Using the definition of the vorticity (Eq. 2.2) and the streamfunction (Eq. 2.4) as well as the continuity equation (Eq. 2.1b) we may derive a Poisson equation relating Ψ to ω as :

$$\nabla^2 \Psi = -\omega . \tag{2.5}$$

Summarizing the above results we may express the mathematical model for the flow in a domain $\mathcal{D}$ by the following set of equations :

$$\boxed{\begin{array}{ll} \dfrac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \nabla^2 \omega & \text{in } \mathcal{D} \\ \nabla^2 \Psi = -\omega & \text{in } \mathcal{D} \\ \mathbf{u} = \nabla \times \boldsymbol{\Psi} & \text{in } \mathcal{D} \\ \omega(\mathbf{x}, 0) = \omega_0(\mathbf{x}) & \text{in } \mathcal{D} \\ \mathbf{u} = \mathbf{U}_b(t) + \Omega(t)\, \widehat{\mathbf{e}_z} \times (\mathbf{x}_s - \mathbf{x}_b) & \text{on } \partial\mathcal{D} \\ \mathbf{u} \to \mathbf{U}_\infty & \text{at } \infty \end{array}} \tag{2.6}$$

The above set of equations is well-posed and is equivalent to the Navier-Stokes equations in two dimensions. The vorticity-streamfunction formulation helps in eliminating the pressure from the unknowns of the equations. However for bounded domains it introduces additional constraints in the kinematics of the flow field and requires the transformation of the velocity boundary conditions to vorticity form. Our numerical method is based on the discretization of the above equations in a Lagrangian frame using particle (vortex) methods.

2.2 Particle (Vortex) Methods

Convection-diffusion equations for an unknown field $q(\mathbf{x},t)$ are expressed in general form as :

$$\frac{\partial q}{\partial t} + \nabla\cdot(\mathbf{c}(\mathbf{x},t)\,q) = \nabla\cdot(d(\mathbf{x},t)\,\nabla q) \tag{2.7}$$

where $\mathbf{c}(\mathbf{x},t)$, $d(\mathbf{x},t)$ are known vector and scalar functions respectively. In particle methods the positions ($\mathbf{x_a}$) of the computational elements are modified according to :

$$\frac{d\mathbf{x_a}}{dt} = \mathbf{c}(\mathbf{x_a},t)$$

so when written in coordinates moving with the particles Eq. 2.7 becomes :

$$\frac{dq}{dt} = \nabla\cdot(d(\mathbf{x},t)\,\nabla q) - q\,\nabla\cdot\mathbf{c}(\mathbf{x},t)$$

The incompressible Navier-Stokes equations when expressed in vorticity form are equations of the above type where $q(\mathbf{x},t) = \omega(\mathbf{x},t)$ is the vorticity and $\mathbf{c}(\mathbf{x},t) = \mathbf{u}(\mathbf{x},t)$ is the velocity field (with $\nabla\cdot\mathbf{u} = 0$) and $d(\mathbf{x},t) = \nu$ is the kinematic viscosity (which is constant throughout the computational domain). After these simplifications Eq. 2.3 is expressed in Lagrangian form by the following set of equations:

$$\boxed{\begin{aligned}\frac{d\mathbf{x_a}}{dt} &= \mathbf{u}(\mathbf{x_a},t)\\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= \nu\,\nabla^2\omega\end{aligned}} \tag{2.8}$$

In the context of particle methods it is desirable to replace the right-hand side of equations Eq. 2.8 by integral operators. These operators are discretized using as quadrature points the locations of the particles so that ultimately Eq. 2.8 is replaced by a set of O.D.E. 's whose solution is equivalent to the solution of the original set of equations

To this effect the velocity field may be determined by the vorticity field using the Green's function formulation for the solution of Poisson's equation (Eq. 2.5).

$$\mathbf{u} = \mathbf{K} * \boldsymbol{\omega} = -\frac{1}{2\pi}\int \mathbf{K}(\mathbf{x}-\mathbf{y})\times\boldsymbol{\omega}\,d\mathbf{y} + \mathbf{U}_0(\mathbf{x},t)$$

where $\mathbf{U}_0(\mathbf{x},t)$ is the solution of the homogeneous Poisson equation for the velocity field and $\mathbf{K}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})/|\mathbf{x}-\mathbf{y}|^2$ is the convolution kernel.

The Laplacian operator may be approximated by an integral operator (Raviart, 1987) as well so that :

$$\nabla^2\omega \approx G * \omega = \int G(|\mathbf{x}-\mathbf{y}|)\,[\omega(\mathbf{x}) - \omega(\mathbf{y})]\,d\mathbf{y}$$

The boundary condition Eq. 2.1d is enforced by formulating the physical mechanism it describes. The solid wall is the source of vorticity that enters the flow. A vorticity flux ($\frac{\partial\omega}{\partial n}$) may be determined on the boundary in a way that ensures Eq. 2.1d is satisfied. It is shown (Ch.5) that this mechanism of vorticity generation can be expressed by an integral operator as well :

$$\frac{\partial\omega}{\partial t} = \int H(\mathbf{x}-\mathbf{y})\,\frac{\partial\omega}{\partial n}(\mathbf{y})\,d\mathbf{y}$$

Using the above integral representations for the right-hand side of Eq. 2.8 we obtain the following set of equations.

$$\boxed{\begin{aligned}\frac{d\mathbf{x_a}}{dt} &= -\frac{1}{2\pi}\int \mathbf{K}(\mathbf{x_a}-\mathbf{y})\times\boldsymbol{\omega}\,d\mathbf{y} + \mathbf{U}_0(\mathbf{x_a},t)\\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= \nu\int G(|\mathbf{x_a}-\mathbf{y}|)\,[\omega(\mathbf{x_a}) - \omega(\mathbf{y})]\,d\mathbf{y}\\ &\quad + \int H(\mathbf{x_a}-\mathbf{y})\,\frac{\partial\omega}{\partial n}(\mathbf{y})\,d\mathbf{y}\end{aligned}} \qquad (2.9)$$

Note that the first of Eq. 2.9 is an exact equation whereas the second is an approximation. For the present numerical scheme the vorticity field is considered as a discrete sum of the vorticity field of the particles, having core radius ϵ, strength $\Gamma(t)$ and the shape of their vorticity field is determined by the function η_ϵ so that :

$$\omega(\mathbf{x},t) = \sum_{n=1}^{N}\Gamma_n(t)\,\eta_\epsilon(\mathbf{x} - \mathbf{x}_n(t))$$

When this expression for the vorticity is substituted in Eq. 2.9 the singular integral operators K,G are convolved with the smooth function η_ϵ and are replaced by smooth operators K_ϵ, G_ϵ. These integrals are subsequently discretized using a quadrature having as quadrature points the locations of the particles. Assuming that each particle occupies a region of area h^2 and that the shape of the body is discretized by M

panels then algorithmically the method may be expressed as :

$$
\begin{aligned}
\frac{d\mathbf{x}_i}{dt} &= -\frac{1}{2\pi}\sum_{j=1}^{N}\Gamma_j\, K_\epsilon(\mathbf{x}_i-\mathbf{x}_j) \;+\; \mathbf{U}_0(\mathbf{x}_i,t) \\
\frac{d\Gamma_i}{dt} &= \nu\sum_{j=1}^{N}[\Gamma_j \;-\; \Gamma_i]\, G_\epsilon(|\mathbf{x}_i-\mathbf{x}_j|) \\
&\quad + \sum_{m=1}^{M} H(\mathbf{x}_i-\mathbf{x}_m)\,\frac{\partial\omega}{\partial n}(\mathbf{x}_m) \\
\Gamma_i(0) &= \omega(\mathbf{x}_i,0)\,h^2 \qquad i \;=\; 1,2,\ldots,N
\end{aligned}
\tag{2.10}
$$

The characteristic of the present method is the replacement of the differential operators by integral operators. The advantage in employing integral operators is based on their stability and efficiency. Integral operators are bounded and smoothing, so that discrete approximations have bounded condition number as the mesh is refined. The representation of the convective terms avoids many difficulties associated with its discretization on an Eulerian mesh such as excess numerical diffusion. However the accuracy of the method relies on the accuracy of the quadrature rule as information needs to be gathered from the possibly scattered particle positions.

2.3 Algorithmic Implementation

The present vortex method is implemented in a time stepping algorithm that proceeds by generating the particle trajectories and appropriately modifying the particle strengths. The simulations of bounded and unbounded flows are distinguished by the enforcement of the boundary condition.

2.3.1 Infinite Domain

For the simulation of flows with no solid boundaries, Eq. 2.10 are integrated simultaneously in time. As it is dictated by Eq. 2.8 **no fractional step** algorithm is necessary as the particle locations and strengths may be considered to be the elements of a common vector of unknowns that are updated simultaneously at each time step.

2.3.2 Bluff Body Flows - A Fractional Step Algorithm

For flows with solid boundaries a difficulty arises by the implicit enforcement of the boundary condition (Eq. 2.1d) in the present formulation, as it is discussed in Ch.5. Eq. 2.10 is not integrated simultaneously in time but instead a fractional step algorithm is employed. The governing equations are solved via a splitting scheme that accommodates the enforcement of the BC.

Let us assume that at the n-th time step (corresponding to time $t - \delta t$) the vorticity field has been computed and we seek to advance the solution to the next time step (time t). The following two step procedure is implemented :

- **Step 1a :** Using as initial conditions $f(\mathbf{x}) = \omega^n(\mathbf{x}^n, n\delta t)$ we solve :

$$\begin{aligned} \omega_t + \mathbf{u}\cdot\nabla\omega &= 0 && \text{in} \quad \mathcal{D}\times[t-\delta t, t] \\ \omega(\mathbf{x}, t-\delta t) &= f(\mathbf{x}) && \text{in} \quad \mathcal{D} \\ \mathbf{u}\cdot\mathbf{n} &= \mathbf{U_s}\cdot\mathbf{n} && \text{on} \quad \partial\mathcal{D}\times[t-\delta t, t] \end{aligned} \tag{2.11a}$$

  As described in section (1.2), particles are advanced via the Biot-Savart law to resolve Eq. 2.11. The no through flow boundary condition ($\mathbf{u}\cdot\mathbf{n} = \mathbf{U_s}\cdot\mathbf{n}$) is necessary to obtain a well-posed problem and is enforced by computing a potential flow correction to the vortical velocity field.

- **Step 1b :** Subsequently the diffusive part of Eq. 2.3 is resolved by the following equation :

$$\begin{aligned} \omega_t - \nu\nabla^2\omega &= 0 && \text{in} \quad \mathcal{D}\times[t-\delta t, t] \\ \omega(\mathbf{x}, t-\delta t) &= f(\mathbf{x}) && \text{in} \quad \mathcal{D} \end{aligned} \tag{2.11b}$$

  Note that no boundary condition is explicitly enforced in this substep The no-slip condition is enforced in the following stage.

  Algorithmically then Step 1 may be expressed as:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= u^n(\mathbf{x}^n, n\delta t) = K\star\omega^n(\mathbf{x}^n, n\delta t) \\ \frac{\partial\omega_1'}{\partial t} &= \nu\nabla^2\omega_1' = G\star\omega^n(\mathbf{x}^n, n\delta t) \end{aligned}$$

At the end of Step 1 a vorticity field ω_1' has been established in the fluid.

- **Step 2a :** The no-slip boundary condition is enforced in this stage by a vorticity (not particle) creation algorithm. The spurious slip velocity that is observed on the surface of the body at the end of Step 1 may be translated to a vorticity flux (see Ch.5).

$$U_{\text{slip}} \rightarrow \frac{\partial \omega}{\partial n} \qquad \text{on } \partial\mathcal{D}$$

- **Step 2b :** The computed vorticity flux generates vorticity in the fluid. The vorticity field is modified by this viscous mechanism as described by the following set of equations :

$$\begin{aligned} \frac{\partial \omega_2'}{\partial t} - \nu \nabla^2 \omega_2' &= 0 && \text{in } \mathcal{D} \times [t-\delta t, t] \\ \omega_2'(\mathbf{x}, t-\delta t) &= 0 && \text{in } \mathcal{D} \\ \frac{\partial \omega_2'}{\partial n} &= F(U_{\text{slip}}(\omega_1')) && \text{on } \partial\mathcal{D} \end{aligned} \tag{2.12}$$

Note that the diffusion equation is solved here with homogeneous initial conditions as the initial vorticity field was taken into account in the previous substep.

The solution at Step 2 is a vorticity field ω_2' which we superimpose onto the solution of Step 1 to obtain the vorticity distribution at the next time step

$$\omega^{n+1} = \omega_1' + \omega_2'$$

2.3.3 A Note on the Enforcement of Boundary Conditions

In a time stepping algorithm at the end of the step the no-slip boundary condition for a stationary body ($\mathbf{u} = 0$) has been enforced. However, this implies the simultaneous enforcement of the no-through flow ($\mathbf{u} \cdot \mathbf{n} = \partial\Psi/\partial s = 0$) and the tangency ($\mathbf{u} \cdot \mathbf{s} = \partial\Psi/\partial n = 0$) boundary condition as well.

This may be seen by extending the definition of the streamfunction in the computational domain to the region occupied by the body. Eq. 2.5 applies in that region and since there is no vorticity inside the body the streamfunction satisfies Laplace's equation ($\nabla^2\Psi = 0$). Hence inside the body ($\mathcal{B}$) :

$$\int_{\mathcal{B}} \Psi \nabla^2 \Psi dA = 0$$

Applying Green's identity to the above equation we obtain

$$\int_{\mathcal{B}} |\nabla \Psi|^2 dA \;=\; -\oint_{\partial \mathcal{B}} \frac{\partial \Psi}{\partial n} \Psi ds$$

Enforcing then the tangency condition is equivalent to requiring that $\Psi = \text{const.}$ and $\partial \Psi / \partial s = 0$ on the body surface as well. So as the end of one time step coincides with the beginning of another enforcing the no-slip condition guarantees that the no-through flow is explicitly enforced at each time step.

In a fractional step algorithm the particles are advanced solving Eq.1.10. When a single step method (such as Adams-Bashforth or Euler) is used there is no need therefore to explicitly enforce the no-through flow condition. However when a substepping time integration is used (such as Runge Kutta) the no-through flow boundary condition must be explicitly enforced during these substeps.

CHAPTER 3

Vectorization of Fast Vortex Methods

Vortex methods, have attracted the interest of many researchers for the study of unsteady incompressible flows at relatively high Reynolds numbers. In such flows the vortical structures, which are the key to describing the whole flow field, tend to be confined in limited regions of the domain and vortex methods inherently adapt to resolve those regions. In this method the vorticity field is described by a set of N particles. The velocity field is computed then at each of the particles, resulting in a set of ODE's that needs to be integrated in time to determine their trajectories. This may alternatively be considered as an N-body problem for the particles. The simplest method for computing the velocities on the particles requires work that is proportional to N^2 as all pairwise interactions need to be computed. This renders the method prohibitive if one wants to use large numbers of particles to resolve all spatial scales in the flow.

The N-body problem appears in a diverse number of scientific fields (astrophysics, plasma physics, etc.) and techniques for the reduction of the computational cost have been addressed in many different perspectives. Solution methods to this problem may be classified in two broad categories: The *hybrid methods* (particle-mesh) and *direct* (particle only) methods. An extensive survey on *hybrid methods* may be found in the book of Hockney and Eastwood (1981). The cost of the hybrid methods is of $\mathcal{O}(\mathrm{N} + \mathrm{M}\log\mathrm{M})$ (where M is the number of mesh points), but their performance degrades when the clustering of the particles is highly non-uniform and they tend to introduce numerical diffusion. The direct methods compute the approximate velocity field as induced by clusters of particles using a certain number of expansions for each cluster. This number is determined by the accuracy required in the solution of the problem and the type of the field that is approximated. A hierarchical (tree) data structure is associated with the particles and is implemented in order to determine when those expansions may be used instead of the exact pairwise interactions to preserve the accuracy of the solution. The tree is used to establish interaction lists for the particles.

Originally the *direct* technique was developed for the simulation of gravitational problems by Barnes and Hut (1986) and Appel (1985) developed a data structure to facilitate such computations. The application of direct schemes in vortex methods has been primarily exploited by Greengard and Rohklin (1987) and Van Dommelen and Rudensteiner (1989) .They may be referred to as the box-box *(BB)* and particle-box *(PB)* algorithms respectively. Their computational cost theoretically scales as N or $N\log N$, respectively, but their applicability and efficient use of available super-computer architectures (vector/parallel) is what really determines the cost of these methods. The issue that arises then in the development of these codes is the question of parallel or vector implementation ? Although this question is usually answered by the availability of resources it seems worth exploring the possibilities. The nature of the algorithm was originally thought to be such that it would lend itself easier to a parallel implementation (Barnes and Hut, 1986) the main constraint being the recursive descent of the data structure. A naive implementation of this algorithm on a vector computer would require that this recursive scheme is unrolled into a sequence of iterative procedures. Such an implementation however would result in inefficient vectorization and no advantage would be taken of the pipelining capabilities.

The parallel implementation of the methods has been addressed at various degrees of sophistication and for a diverse number of problems (see for example Pépin (1990), Katzenelson (1989) and Salmon (1991)). Vectorization of the method has also been successfully addressed in works by Van Dommelen and Rudensteiner (1989) for vortex methods using the *PB* algorithm. Recently (1990) Hernquist, Makino, and Barnes published a series of papers demonstrating that the tree descent can indeed be vectorized. They applied their strategy to the Barnes - Hut algorithm obtaining orders of magnitude increase in the computational speed of the algorithm. Usually the most time consuming part of these schemes has been the building and descending of the tree in order to establish interaction lists. Once these have been established the velocity field of the particles may easily be determined. The vectorization of the *BB* algorithm for vortex methods is addressed in this work and its efficiency is compared to that of a *PB* scheme.

A data structure is implemented which lends itself to vectorization, in the construction and descent level, combining some of the ideas presented in the works of Barnes, Hernquist and Makino. It is applied to both (*PB* and *BB*) algorithms and comparisons are made as to what efficiencies may be obtained. The timings compare favorably with those reported in the past (Van Dommelen and Rudensteine,

1989) and it is demonstrated that an efficient implementation of the BB scheme can outperform the PB one for numbers of particles more than a few thousand.

3.1 Multipole Methods for the Velocity Evaluation

In the context of vortex methods the vorticity field is discretized by a set of elementary vortices whose individual field is expressed by a cut-off function $\phi(x)$. The superposition of these fields determines the vorticity ω, at any location $\mathbf{x}$, as :

$$\omega(\mathbf{x}) = \sum_{n=1}^{N} \Gamma_n \phi(\mathbf{x} - \mathbf{x}_n) \tag{3.1}$$

If $\phi(\mathbf{x})$ is a δ-function or we are interested in the velocity field (u, v) in a location (x, y) away from the particles, the velocity field is expressed in complex form as:

$$\boxed{V(Z) = \frac{i}{2\pi} \sum_{n=1}^{N} \frac{\Gamma_n}{Z - Z_n}} \tag{3.2A}$$

where $Z = x + iy$ and $V = u - iv$. In order to compute the velocity at each one of the particles the above sum implies $\mathcal{O}(N^2)$ operations. To reduce this computational cost the geometrical distribution of the vortices is exploited. The key observation is that the velocity induced by a cluster of particles need not be computed directly from its individual members. Instead the velocity field induced by a group of M particles clustered around a centre Z_M may be approximated by a finite number (P) of multipole expansions. At distances greater than the radius R_M of this cluster this approximation converges geometrically. This is the basis for the PB scheme. The BB scheme introduces one more step. The expansions of a certain cluster may be translated and computed with the desired accuracy at the center of another cluster. Subsequently those expansions are used to determine the velocity of the particles in the second cluster. The derivation of those expansions is based on the following two identities of complex numbers :

$$\frac{1}{1 - z} = \sum_{n=0}^{\infty} z^n; \qquad \text{for } |z| \leq 1 \tag{$*$}$$

$$\sum_{k=0}^{P} \alpha_k (Z - Z_0)^k = \sum_{l=0}^{P} \Big(\sum_{k=l}^{P} \alpha_k \binom{k}{l} (-Z_0)^{k-l} \Big) Z^l. \tag{$**$}$$

Adding and subtracting Z_M on the denominator of Eq. 3.2A the velocity field induced by the cluster may be expressed as :

$$V(Z) = \frac{i}{2\pi} \frac{1}{Z - Z_M} \sum_{m=1}^{M} \frac{\Gamma_m}{1 + (Z_M - Z_m)/(Z - Z_M)}$$

By expanding now the denominator inside the sum using $(*)$ and keeping (P) terms in the expansions the velocity is determined by :

$$\boxed{V(Z) = \frac{i}{2\pi} \frac{1}{Z - Z_M} \sum_{k=0}^{P} \frac{\alpha_k}{(Z - Z_M)^k}} \qquad (3.2B)$$

The complex coefficients α_k express the moments of the discrete vorticity distribution in the cell and are computed by :

$$\boxed{\begin{aligned} \alpha_k &= \sum_{m=1}^{M} \Gamma_m (Z_m - Z_M)^k \\ k &= 0, \ldots, P \end{aligned}} \qquad (3.2C)$$

To make the computations more efficient the coefficients of boxes that belong to coarser levels of the hierarchy are not constructed directly from the particles. Instead they are obtained by a shifting of the expansions of their descendants. To obtain then the expansions of a parent box from those of its children we use identity $(**)$ to get :

$$\boxed{\alpha_l^{\text{parent}} = \sum_{k=0}^{l} \binom{l}{k} \alpha_k^{\text{children}} (Z_M^{\text{children}} - Z_M^{\text{parent}})^{l-k}} \qquad (3.2D)$$

We may observe now from Eq. 3.2B, that at a distance R from the center of the cluster the rate of convergence of the expansions would be proportional to $(R_M/R)^{P+1}$ and, for example, if $R \geq 2R_M$ this rate is proportional to $(1/2)^{P+1}$, implying the geometric convergence of the series.

The above expansions are the main tools for the *PB* algorithm. However one may proceed one step further and recast the Laurent series into Taylor series and consider the interactions between groups of particles. These interactions take place in the form of shifting the center of expansions of one cluster (M) to the center of another (G). Those expansions are then used to compute the velocity of the particles in each box. This results in eliminating the logN factor from the work count of the scheme.

To obtain these expansions we add and subtract Z_G in the denominator of the expression Eq. 3.2A so we have that the velocity induced by the group M at a point Z in the neighborhood of Z_G is equal to :

$$V(Z) = \frac{i}{2\pi} \sum_{k=0}^{P} \frac{\alpha_k}{(Z_G - Z_M)^{k+1}(1+\xi)^{k+1}}$$

where $\xi = (Z - Z_G)/(Z_G - Z_M)$. Expanding this expression for ξ using $(*)$, the velocity field induced by particles in the cluster M, at a point Z in the neighborhood of Z_G is:

$$\boxed{V(Z) = \frac{i}{2\pi} \sum_{l=1}^{P} \delta_l (Z - Z_G)^l} \qquad (3.2E)$$

where the coefficients δ_l are determined by :

$$\boxed{\delta_l = \frac{(-1)^{l+1}}{(Z_G - Z_M)^l} \sum_{k=0}^{P} \binom{l+k-1}{k} \frac{\alpha_k}{(Z_G - Z_M)^k} \qquad l = 1, \ldots, P} \qquad (3.2F)$$

Interactions are computed at the coarsest possible level while satisfying the accuracy criterion. However the velocities of the individual particles are computed from the expansions of the finest boxes in the hierarchy. Hence the expansions of the coarser level boxes have to be transferred down to their descendants as follows :

$$\boxed{\delta_l^{\text{children}} = \sum_{k=l}^{P} \binom{k-1}{k-l} \delta_k^{\text{parent}} (Z_M^{\text{children}} - Z_M^{\text{parent}})^{k-l}} \qquad (3.2G)$$

This shifting uses (**) and, as in Eq. 3.2C, does not introduce any additional errors. One may observe here the different use of the coefficients α_k and δ_l when computing velocities on the particles. The coefficients δ_l of a certain box are used to compute the velocity on the particles of the same (childless) box whereas the α_k's are used to compute the velocities due to particles that belong to well separated boxes.

Proofs for the above results may be found in a more rigorous and complete form in Greengard and Rohklin (1987). They show that the series converges if the distance between interacting boxes and/or particles is at least twice as large as the radius of the cluster involved. By using the expansions when clusters are separated by larger distances one may reduce the number of expansions that are necessary to obtain the same level of accuracy. This trade-off may be optimized by linking the number of expansions employed to the distance between the interacting pairs dynamically (Salmon, 1991). However this may result in increased cost for the construction of the interaction lists and it would complicate the algorithm and reduce its vectorizability.

3.2 The Data Structure

The two-dimensional space is considered to be a square enclosing all computational elements. We apply the operation of continuously subdividing a square into four identical squares until each square has only a certain maximum number of particles in it (see Fig.3.1 for an example) or the maximum allowable level of subdivisions has been reached (the latter requirement seems obligatory when one programs in FORTRAN and has to predefine array dimensions). This procedure for a roughly uniform particle distribution results in $\mathcal{O}(\log_4 \mathrm{N})$ levels of squares.

The two fast algorithms under discussion, *PB* and *BB*, exploit the topology of the computational domain each with a different degree of complexity and efficiency. The hierarchy of boxes defines a tree data structure which is common for both algorithms. However its key ingredients and address arrays are implemented in a different way as explained below for the two algorithms. The tree construction proceeds level by level starting at the finest level of the particles and proceeding upwards to coarser box levels.

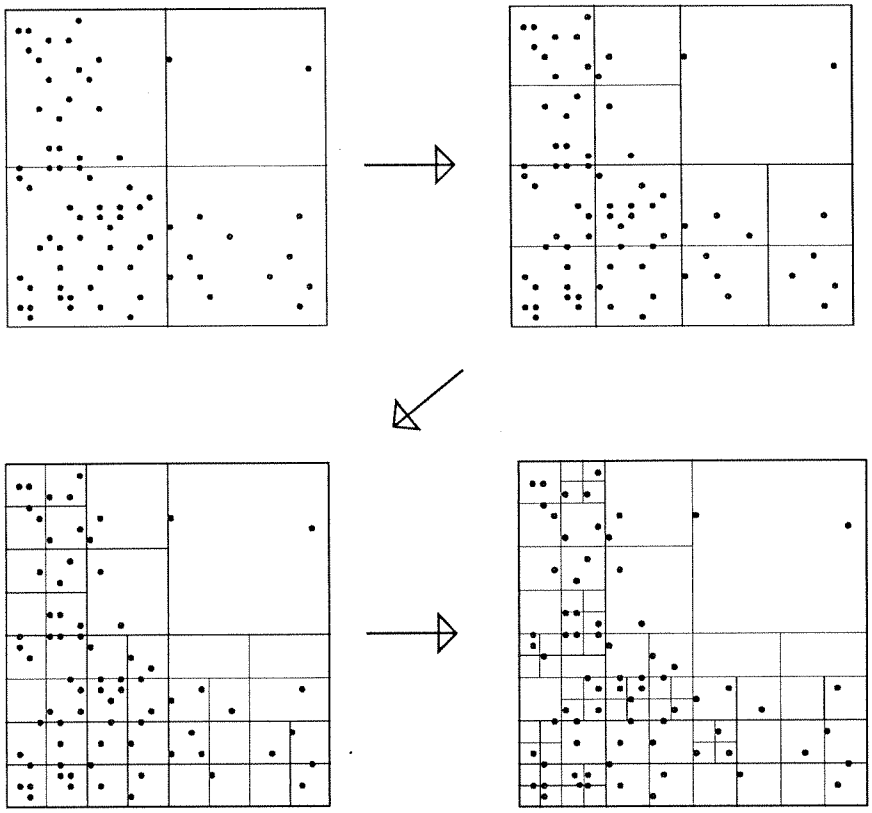


FIG. 3.1 An example showing four stages of subdividing the computational domain. In the final stage each box contains less than three particles.

Due to the simplicity of the geometry of the computational domain the addressing of the elements of the data structure is facilitated significantly. As the construction proceeds pointers are assigned to the boxes so that there is direct addressing of the first and last particle index in them as well as there is direct access to their children and parents. This facilitates the computation of the expansion coefficients of the children from the expansions of the parents for the *BB* algorithm and the expansions of the parents from those of the children for the *PB* algorithm.

3.2.1 Parameters of the Data Structure

The data structure is used to determine when the expansions are to be used and when pairwise interactions have to be calculated. Usually this data structure is referred to as the '(family) tree ' of the particles. It helps in communicating to the computer the geometric distribution of the particles in the computational domain. The particles reside at the finest level of the structure. Clusters of particles form the interior nodes of the tree and hierarchical relations are established. The data structure adds to the otherwise minimal memory requirements of the vortex method. This extra memory however is the tradeoff for the speed of the fast algorithms. One may add several features to this data structure trying to relate to the computer architecture as much information as possible for the particle configuration. However these memory requirements have to maintain a certain degree of uniformity for all the levels of the tree. So if one wants to use large numbers of particles (hence several levels of subdivisions) these memory requirements should be kept to a reasonable level.

The formation and descent of the tree add to the cost of the algorithms and a non-refined implementation may result in the degradation of the whole algorithm. A data structure may be Lagrangian or Eulerian. The former adapts automatically to the locations of the particles and can be the same for several steps. The latter has to be reconstructed at every step as the particles change positions in the domain. There are several ways that nearby particles could be clustered together and some of the decisions to be made are :

The center of expansions. This can be either the geometric center of a certain region in space or the center of mass of the distribution of particles or some other location chosen so that the data structure is conveniently addressed and the expansions converge rapidly. Choosing the center of 'mass' implies that for a uniform particle distribution and a certain number of particles per cluster the minimum radius for convergence is expected. On the other hand the geometric center (as in the present work) facilitates tremendously the addressing of the boxes in the data structure.

The cluster size. Accuracy requirements impose that the cluster size should be as small as possible while efficiency considerations dictate that clusters should contain enough particles so that the use of the expansions is beneficial. There are different approaches to that as others require the finest clusters to contain fewer than

L_{min} (e.g. for the Barnes-Hut algorithm $L_{min} = 2$) particles where others impose the finest level of subdivision beyond which no further subdivisions take place. The former procedure seems to offer more accurate expansions whereas the latter economizes in memory and compactness of the data structure especially when more than one particle resides at the finer boxes and large numbers of terms are to be kept in the expansions. In the present algorithms we follow a hybrid strategy as we keep at least L_{min} particles per box until we reach a predetermined finest level of boxes. The number L_{min} may be chosen by the user depending on the particle population and configuration so as to achieve an optimal computational cost.

A related issue is how one subdivides a cluster or box with more than L_{min} particles. Practical implementations of the method resort basically to two techniques. One may split the parent cluster into two children resulting in a so called binary tree. The direction of this dissection depends on the distribution of the particles, attempting to optimally adapt the data structure to the locations of the particles in the computational domain. In practice this has the benefit of requiring fewer terms in the expansions to obtain a certain accuracy (for a relatively uniform particle distribution). Alternatively, as in the present scheme, one may simply divide the box into four boxes. This would require a larger number of expansions to be calculated and stored to obtain the required accuracy, as the radius for convergence is not minimized. However these 'extra' terms have a minimal effect to the overall cost of the scheme as they appear in fully vectorized parts of the algorithm. Moreover the regularity of such a procedure facilitates the logic of the algorithm and the construction and addressing of the data structure.

Addressing the clusters. A key factor in the computer implementation of the method, addressing should be such that it does not inhibit the vectorization or the parallel implementation of the method. Traversing and building the hierarchical tree is highly dependent on this procedure. A simpler technique would be to store information (such as geometric location, size, family ties etc.) for the tree nodes in the memory. Such a procedure may severely limit the number of particles that can be computed (Pépin,1990) if excess information is stored. Besides it would degrade the performance for machines (such as the CRAY-2) where memory access is computationaly intensive. However both restrictions do not seem to consist major drawbacks in the present implementation on the CRAY YMP, provided a reasonable number of subdivisions (less than 10) is allowed.

Another key issue is the addressing of the particles. As particles are usually associated with a certain box it is efficient to sort the particle locations in the memory so that *particles that belong to the same box occupy adjacent locations in the memory devoted to the particle arrays.* Such memory allocation enhances the vectorization tremendously as very often we loop over particles of the same box (e.g., to construct the expansions at the finest level, or to compute interactions) and the loops have an optimum stride of one.

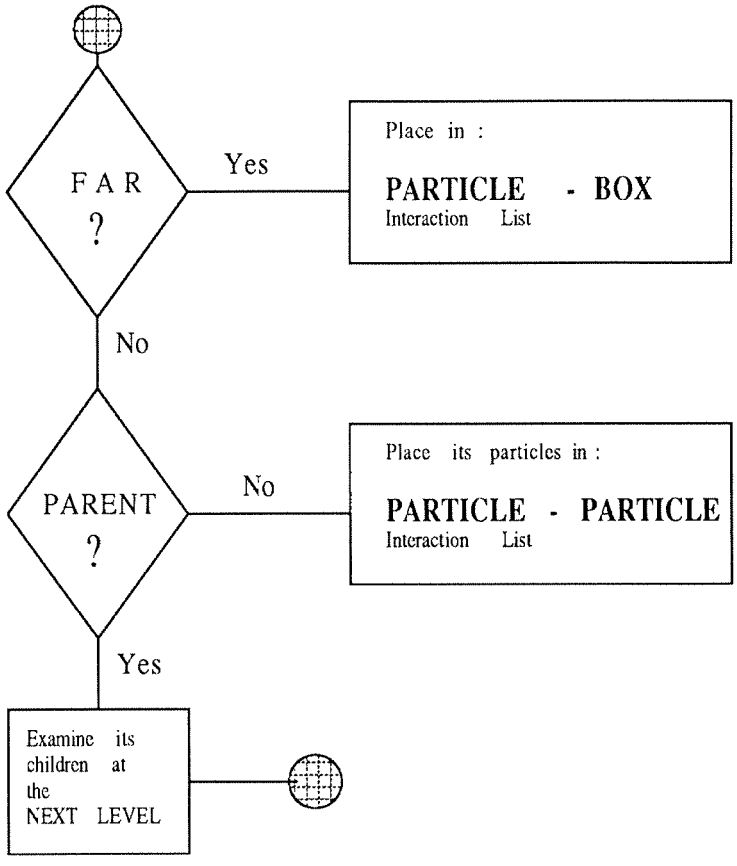


FIG. 3.2 Flow Chart for the tree traversal of the *PB* algorithm for *all particles of a childless box.*

As one can deduce, the possible combinations of the above features may result in a number of different implementations. Depending on the implementation a different degree of vectorization may then be achieved.

3.3 Description of the Algorithms

In both algorithms, described herein, we may distinguish three stages:

- Building the data structure (tree)
- Establishing the interaction lists (by non-recursively descending the tree)
- Velocity evaluations for all particles in the domain.

The building of the data structure is common for both algorithms but they differ in the tree descent and the velocity evaluation. Basic requirements for an efficiently vectorized code are the simplicity of the algorithm, the existence of long vectorizable loops (i.e., simple loops, odd stride e.t.c) and the reduction of memory referencing. Care has been exercised at all stages so that the maximum degree of vectorization is achieved and efficient calculations result. In the present implementation the building of the data structure consumes about $5-7\%$ of the time whereas the descent consumes another $5-10\%$ so that the largest amount is spent in computing the velocities.

3.3.1 The Particle-Box Algorithm

The hierarchical structure of the algorithm has a logical complexity that implies a recursive procedure. The algorithm may be easily described by the following code :

subroutine interact (n,C)

IF (particle **n** is well separated from cell **C**)

velocity = sum of expansions of cell **C** on **n**

ELSE

CALL **interact (n,children of C)**

ENDIF

return

Note the recursiveness of this subroutine in the ELSE block. A straightforward approach to the non-recursive programming of this subroutine, in an attempt to vectorize it, would be to unroll it. Such a procedure would introduce a depth-first search of the tree. However this is not very efficient because for most applications the depth of the tree is not long enough to enable optimized vector operations. As reported by Makino (1990), Barnes introduced interaction lists associated with each particle and then vectorized by looping over the particles. Every particle had its own interaction list but increased memory referencing and the additional complexity of the algorithm resulted in degradation of the performance. In our algorithm we employ the following alternate procedure:

Step 1 : Building the data structure (tree)

Step 1a : For each of the squares at each level that are not further subdivided we compute the p-terms of the multipole expansions. These expansions are used to describe the influence of the particles at locations that are well separated from their cluster. The *cost* of this step is $\mathcal{O}(Np)$.

Step 1b : The expansions of all parent boxes are constructed by shifting the expansion coefficients of their children. The tree is traversed upwards in this stage. Rather than constructing the expansions of all the members of a family (that is traverse each branch until the root is reached) we *construct the expansions of all parent boxes at each level simultaneously.* This enables long loops over the parent boxes at each level. Care is taken so that the procedure is fully vectorized by taking advantage of the regularity of the data structure and the addressing of the boxes in the memory. Moreover the regularity of the data structure allows us to precompute many coefficients that are necessary for the expansions. The cost of this step is $\mathcal{O}(Np^2)$, as each shifting requires p^2 operations and there are at most $(4N-1)/3$ boxes in the computational domain.

Step 2 : Establishing of interaction lists

In the present algorithm a **breadth-first** search is performed at each level to establish the interaction lists of each particle (cell). This search is facilitated by the regularity of the data structure and the identification arrays of the cells in the tree. At each level interaction lists are established for the particles (cells) by looping across the cells of a certain level.

Algorithmically this operation is represented as follows:

subroutine setlist (n)
Do 1 l = 1, levelmax
    CALL FARCLOSE(Z(n),ds(l),IZ(l),kxm,Lstxm,kfr,Lstfr,kcls,Lstcls)
    CALL CLOSECHECK(kcls,Lstcls,kpp,Lstpp,kxm,Lstxm)
    CALL GATHER(kfr,Lstfr,kpp,Lstpp,ZP,GP,ZB,EB)
1 Continue
return

For each particle **n** the hierarchy of cells is traversed breadth first. The check for faraway or close boxes is performed within the subroutine FARCLOSE that is fully vectorized. Initially the addresses of `kxm` cells that belong to the coarsest level are stored in array `Lstxm`. By checking the locations of the boxes at the grid of level `l` and the respective location of the particle on that grid the cells are identified as either faraway (stored in `Lstfr`) or nearby (stored in `Lstcls`). Boxes that are far interact with the particle and are placed in a P-B interaction list `Lstpp` along with their expansions `EB` and locations `ZB` in subroutine GATHER. Boxes that are close however are further examined in the fully vectorized subroutine CLOSECHECK. Those that are childless have their particles stored in an array `Lstpp` so that they interact directly. Those that are parents have their children stored in array `Lstxm` so that they are fed back in FARCLOSE at the finer level. Note that at the finest level all boxes are considered childless and subroutine CLOSECHECK need not be called. This way the interaction lists for the particles are formed successively and a fully vectorized force calculation calculates the velocities of the particles at the following stage.

Note now that this depth first search for interaction lists is further facilitated by the following observation. Every particle belongs to a childless box. It is easy then to observe that *all particles in the same box share the same interaction list comprised of members of the tree that belong to coarser levels.* This way the tree is traversed upwards for all particles in a childless box together and downwards separately for each particle. It is evident that this procedure is more efficient for uniformly clustered configurations of particles as there would be more particles that belong to childless boxes at the finest level.

The *cost* of this step scales as $\mathcal{O}((N/L_{\min})\log N)$ as there are $\log N$ levels of boxes

and the tree is traversed $(N/L_{\min})$ times.

Step 3 : Computation of the interactions.

Once the interaction lists have been established the velocities of the particles are computed by looping over the elements of the lists. For particles that have the same boxes in their interaction list this is performed simultaneously so that memory referencing is minimized. Moreover by systematically traversing the tree the particle-particle interactions are made symmetric so that the cost of this computation is halfed. Care has been exercised to compute the velocities with the minimum possible number of operations (Section 3.4).

The *cost* of this step is $\mathcal{O}(Np)$.

3.3.2 The Box-Box Algorithm.

This scheme is similar to the *PB* scheme except that here every node of the tree assumes the role of a particle. In other words interactions are not limited to particle-particle and particle-box but interactions between boxes are considered as well. Those interactions are in the form of shifting the expansion coefficients of one box into another and the interaction lists are established with respect to the locations of every node of the tree.

The scheme distinguishes five categories of interacting elements of the tree with respect to a cell denoted by **c**.

- **List 1 :** All childless boxes neighboring **c**.
- **List 2 :** Children of colleagues of b's parents that are well separated from **c**. All such boxes belong to the same level with **c**.
- **List 3 :** Descendants (not only children) of **c's** colleagues (boxes of the same size as **c**), whose parents are adjacent to **c** but are not adjacent to **c** themselves. All such boxes belong to finer levels.
- **List 4 :** All boxes such that box **c** belongs to *their* List 3. All such boxes are

childless and belong to coarser levels.

- **List 5 :** All boxes well separated from **c's** parents. Boxes in this category do not interact directly with the cell **c**.

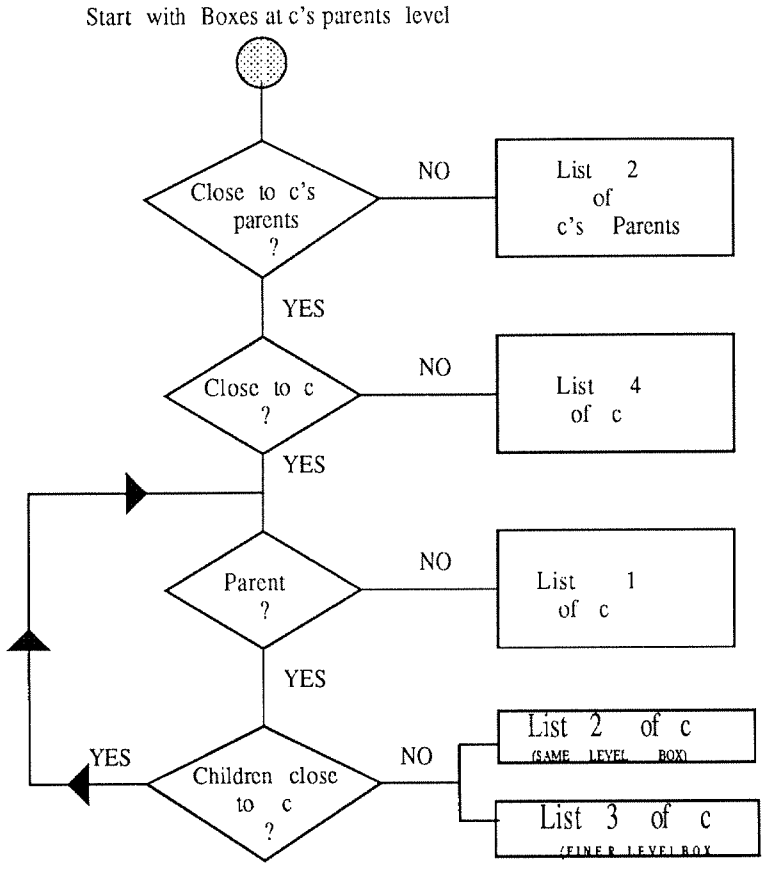


FIG. 3.22 Flow Chart for the Tree Traversal of the *BB* scheme for a *cell* **c**

If the cell **c** is childless it may have interacting pairs that belong to all four lists. However if it is a parent it is associated with boxes that belong to lists 2 and 4 as described above. These observations are directly applied in our algorithm and we may distinguish again the following 4 steps.

Step 1 : Building the data structure.

This procedure is the same as for the *PB* scheme. This fact enables us actually to compare directly the two algorithms and asses their efficiency.

Step 2 : Construction of Interaction Lists.

To establish the interaction lists we proceed again level by level starting at the coarsest level. For each level we distinguish childless and parent boxes. In establishing lists 1 and 3 we need only loop over childless boxes whereas to establish lists 2 and 4 we loop over all cells that are active in a certain level.

Step 2a : Here we establish lists 1 and 3. We start at the level of the parents of box **c** and we proceed level by level examining again breadth first, until we reach the finest level of the structure (the particles). The elements of lists 1 are basically the particles and account for the particle-particle interactions. Care is exercised so that this computation is symmetric and we need to traverse the tree downwards only. The elements of List 3 are the boxes and are accountable for the particle-box interactions in this scheme.

More specifically an algorithmic description of our algorithm is given by :

subroutine Lists1&3(lpr,kchlds)

```
Do 1 k = 1,kchlds
    Find the colleagues of k's parents (Lclg)
    Place the children of (Lclg) boxes in (Lstxm)
    Do 2 l = lpr,lmax
        CALL FARCLOSE(ZC,ds(l),Kxm,Lstxm,k3,Lst3,kcls,Lstcls)
        CALL CLOSECHECK(kcls,Lstcls,k1,Lst1,kxm,Lstxm)
    2 Continue
1 Continue
return
```

The outer loop here is over all childless boxes. The colleagues of their parents are identified (either by being retrieved directly from an array or by using FARCLOSE) and subsequently their children are placed in `Lstxm`. Subsequently a call to FARCLOSE distinguishes between faraway and nearby boxes at different levels. Boxes placed in `Lst1` are to interact directly whereas those placed in `Lst3` are to interact as particle-box.

Step 2b : Here we establish interaction lists 2 and 4 for all boxes in the hierarchy. We start at the coarsest possible level and proceed downward until reaching the

level of box **c** to establish the interaction lists. To do so for a certain box we start by examining boxes that are not well separated from its parents (otherwise they would have been dealt with at the coarser level). Subsequently the children of those boxes are examined to establish interaction lists. Algorithmically this operation is represented by:

subroutine Lists2&4(lpr,kbox)

Do 1 k = 1,kbox

Do 2 l = l0,lpr

Boxes close to parents (?)(use FARCLOSE)

Close to parents - Childless(?)(use CLOSECHECK)

Close to parents and Childless - Close to box (?)(Use FARCLOSE)

2 Continue

1 Continue

return

Step 3 : Computations of the interactions

In this scheme we consider three kinds of interactions: the box-box, particle-box and particle-particle interactions. The latter two categories were discussed in the previous section. For the box-box interactions once the respective interaction lists have been established (with members of lists 2 and 4) we need to transfer those expansions down to the ones of the children and add them to the existing ones. This procedure is vectorized by looping over the number of boxes at each level. The use of pointers to access the children of each box enhances this vectorization. Note that an arbitrarily high number of expansions can be calculated efficiently by unrolling the loop over the number of expansions into the previously mentioned loop.

3.4 Practical Formulas for Velocity Calculations

Once the interaction lists have been established the velocities on the particle locations need to be calculated using formulas 3.2(A-F). Because this is the most time consuming part of the algorithm an effort is made to reduce as much as possible the computational cost. A possible increase in the computational speed may be obtained by some ASSEMBLY programming for the velocity evaluations. Following are some of the formulas employed to reduce this computational cost.

▷ *Particle-Particle Interactions :* In order to achieve higher convergence rates in the vortex methods more complicated functions than the delta functions are usually necessary for the description of the vorticity field. This will reduce the efficiency of the algorithm but to minimize the extra cost one could make use of look up tables. In these look up tables in order to get the accuracy required we should not only store the values of the function but its derivatives as well. Using the MATH77 scientific libraries of the CRAY would help reduce this computational cost.

▷ *Particle-Box Interactions :* Formula 3.2B may be calculated using recursive relations. By computing separately real and imaginary parts for the velocities we have that for a particle located at $z = x + i\,y$ the contribution from a box located at $X_M = X_M + i\,Y_M$ having expansions $\alpha_k = \lambda_k + i\,\mu_k$ may be computed with the following algorithm.

$$r_0 = \frac{x - X_M}{(x - X_M)^2 + (y - Y_M)^2} \quad , \qquad f_0 = \frac{y - Y_M}{(x - X_M)^2 + (y - Y_M)^2},$$

$$r_k = r_{k-1} \cdot r_0 - f_{k-1} \cdot f_0$$
$$f_k = r_{k-1} \cdot f_0 + f_{k-1} \cdot r_0$$

and the velocities (u,v) would be given as :

$$u = -\sum_{k=0}^{k=P} \lambda_k \cdot r_k - \mu_k \cdot f_k$$
$$v = \sum_{k=0}^{k=P} \lambda_k \cdot f_k + \mu_k \cdot r_k$$

▷ *Parent Cell Expansions :* The expansions of parent cells are computed from the expansions of their children. This procedure does not introduce any errors and it helps in economizing computer time compared to a direct calculation using the particle locations. The amount of necessary calculations is reduced by exploring the regularity of the location of the children boxes with respect to the parents. Thus in formula 3.2D we have

$$Z = Z_M^{\text{children}} - Z_M^{\text{parent}} = \frac{d}{2} \begin{cases} -1 + i, & \text{for Box 1;} \\ -1 - i, & \text{for Box 2;} \\ 1 - i, & \text{for Box 3;} \\ 1 + i, & \text{for Box 4.} \end{cases}$$

where d is the size of the child cell. Hence we may calculate explicitly the powers $\binom{l}{k} Z^{l-k}$ for all values of l,k thus speeding up the computations inside the loop for the boxes.

▷ *Cell-Cell Interactions:* A procedure similar to the one for the particle-cell interactions is followed here. If $\delta_l = \eta_l + i\theta_l$ are the coefficients of a box located at $Z_G = X_G + iY_G$ which are contributed by a box located at $Z_M = X_M + iY_M$ then these coefficients are computed with the following scheme :

$$r_0 = \frac{X_G - X_M}{(X_G - X_M)^2 + (Y_G - Y_M)^2} \quad , \qquad f_0 = \frac{Y_G - Y_M}{(X_G - X_M)^2 + (Y_G - Y_M)^2},$$

$$\begin{aligned} r_k &= r_{k-1} \cdot r_0 - f_{k-1} \cdot f_0 \\ f_k &= r_{k-1} \cdot f_0 + f_{k-1} \cdot r_0 \end{aligned}$$

and finally:

$$\begin{aligned} R_l &= -\sum_{k=0}^{k=P} \lambda_k \cdot r_k - \mu_k \cdot f_k \\ F_l &= \sum_{k=0}^{k=P} \lambda_k \cdot f_k + \mu_k \cdot r_k \\ \eta_l &= r_l \cdot R_l - f_l \cdot F_l \\ f_k &= r_l \cdot F_l + f_l \cdot R_l \end{aligned}$$

▷ *Children Expansions :* Cell-Cell interactions are performed at the coarsest possible level. Then the expansions of the parent boxes are transferred down to the expansions of their children and are added to the existing ones until the finest level at any branch of the tree is reached. Those expansion coefficients are computed using formula 3.2G in a similar way as for the expansions of parent cells discussed above.

▷ *Cell - Particle Interactions :* Once the expansions of all childless boxes have been obtained they are used to compute the velocities on the particles. Formula 3.2E is used and a procedure similar to the one discussed above is followed. Vectorization is achieved by looping over all particles in a certain childless box. This may be inefficient if only a small number of particles is contained in this box.

3.5 Performance of the Algorithms

Comparisons were made for the two fast algorithms and the $\mathcal{O}(N^2)$ in terms of efficiency and computational speed. We require that the fast algorithms produce the same results as the N^2 scheme with minimum accuracy of six significant figures in the velocity field of a random uniform distribution of particles in a square. We allow up to eight levels in the hierarchy of the boxes and use ten terms in the multipole expansions at all levels.

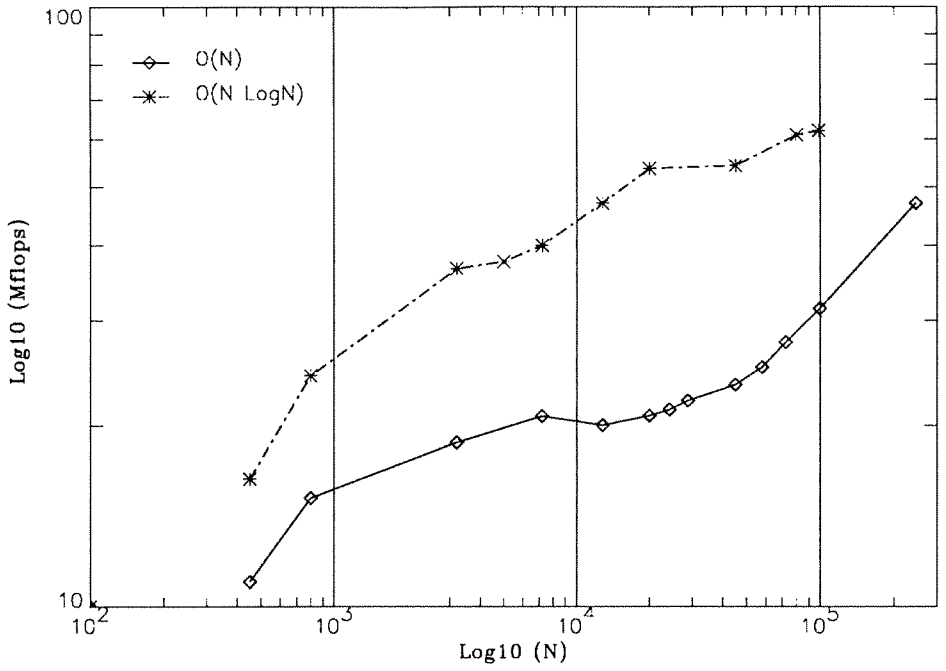


FIG. 3.3 Performance in Mflops of the *PB* and *BB* algorithms on the CRAY/XMP-18

In Fig. 3.3 we show the efficiency in the vectorization achieved for the fast algorithms on the CRAY/XMP-18. The *PB* algorithm is much more efficient than its *BB* competitor in regards with vectorization. This is a result that was expected because of the simplicity of the PP scheme versus the algorithmic complexity of the BB scheme.

This is not the whole story however, since in applications we are mainly interested in the overall speed of the algorithm and not necessarily in its Mflps efficiency. Fig. 3.4 shows the CPU time required on a single processor of an XMP-18 in each of the three methods for an evaluation of all N velocities for N elements with a minimum accuracy

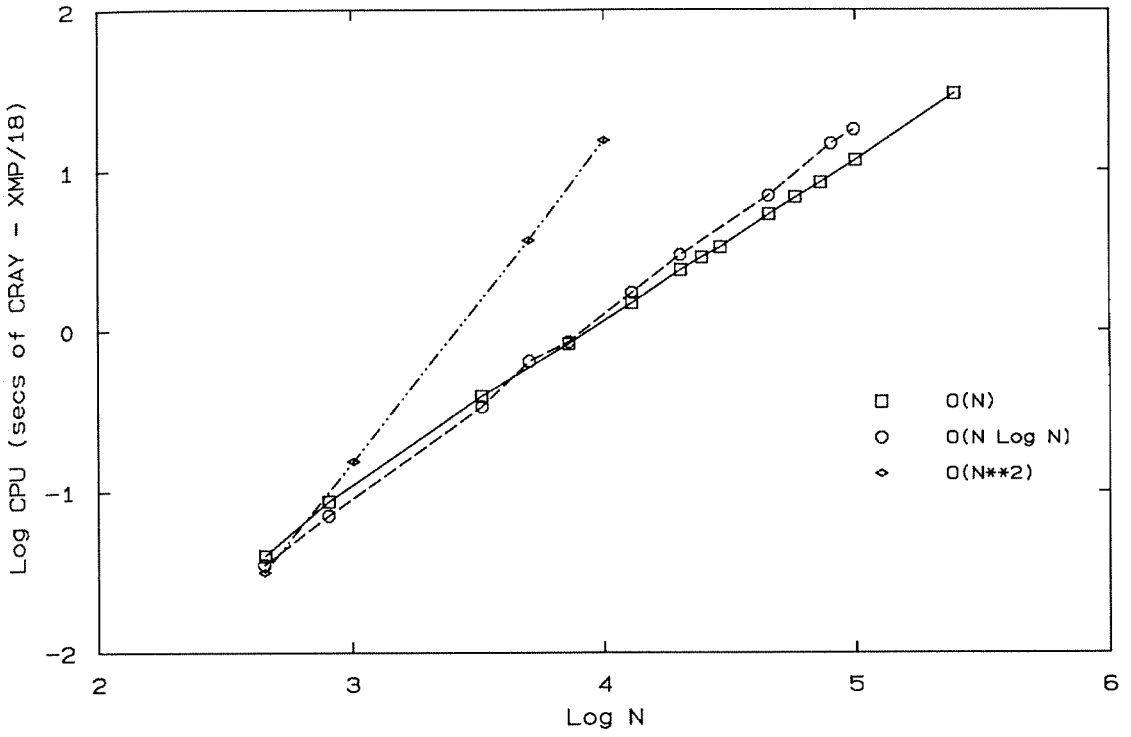


FIG. 3.4 Computational Cost of the PB, BB and direct summation algorithms

of 10^{-6}. Note that fast methods become advantageous for computations involving only $N \approx 500$ particles with the BB method breaking even with the PB method for $N \approx 4400$. Note that the CPU time required on a YMP processor is about 2/3 of the time required on an XMP. A timestep requiring one evaluation of all velocities for a million particles requires about one minute on single processor of a CRAY YMP while the N^2 algorithm would require roughly 24 hours.

CHAPTER 4

Diffusion Schemes for Vortex Methods

Lagrangian methods are numerical schemes well suited for simulations of physical problems governed by hyperbolic equations. In incompressible fluid dynamics a class of Lagrangian methods, the vortex method, is an efficient tool for the study of inviscid flows. The dynamics of such flows are determined by the evolution of their vorticity field and vortex methods are designed so that the computational elements automatically adapt to resolve the regions containing vorticity, without resorting to averaging and smoothing introduced by an Eulerian mesh. However in order to solve the full Navier-Stokes equations the method needs to be enhanced to account for viscous effects. The method is faced with the challenge of approximating the smooth diffusion operator with information gathered from the distorted (due to convection) Lagrangian elements.

We may distinguish three broad categories of Lagrangian methods used for the simulation of viscous incompressible flows. *Moving grid* methods, *pure particle* methods, and the *hybrid* methods. As to *moving grid* methods we refer to the moving-point method (Farmer, 1985) and the free Lagrange method (Fritts, 1985). Methods of this type are close relatives of finite difference schemes distinguished from them by the fact that the mesh is not fixed but it is changing in time due to the convective motion of the points. The diffusion equation is then solved by appropriately integrating the equations using an integral (averaging) approximation of the vorticity and its Laplacian on this mesh. Although these methods are quite popular for the simulation of compressible flows they have not been used extensively in the simulations of incompressible flows. For more details the reader is referred to the works cited above.

Particle (vortex) methods resort to the replacement of the governing differential equations by integral operators that are subsequently discretized using as quadrature points the locations of the particles. The convection equation is solved by following the trajectories of the particles. The diffusion equation is solved by employing its solution in the integral form (Friedmann, 1964). This integral form may be interpreted

in a number of different ways resulting in various numerical schemes. Chorin (1973) allows the particles to undergo a random walk so that their locations when sampled statistically approximate this integral. Leonard (1980) observed that individual particles having a Gaussian core are exact solutions of the diffusion equation and one may use a superposition of their solutions to simulate viscous effects. Raviart (1987) and his coworkers (Cottet, Huberson, Mas-Gallic, (1987)) developed a method (here called the method of Particle Strength Exchange) that approximates the Laplacian by an integral operator that is subsequently discretized on the particles.

Algorithms that use only particles are ususally hampered by the fact that the computational elements become too distorted (due to convection) to accurately approximate the diffusion equation. *Hybrid* methods attempt to alleviate this problem by employing a grid along with the particles. A viscous splitting algorithm is employed where the convection is solved by advancing the particles and the diffusion equation is solved on the grid. An intermediate step is used to couple the two steps by transferring information between the grid and the particles (Jolles and Huberson, 1989) An alternative hybrid scheme employs a domain decomposition technique (Cottet, 1990) where a grid is used to solve the Navier-Stokes equations where viscous effects are important and a particle method is employed to solve the Euler equations where the flow is dominated by convection.

In the present scheme particles are used for the solution of the convection and diffusion equation without resorting to a viscous splitting algorithm (for unbounded flows). The particles are convected using the Biot-Savart law to compute their velocities and the scheme of Particle Strength Exchange (PSE) to account for the viscous effects. Accurate computations are obtained for large times by not allowing the Lagrangian grid to become too distorted as the particles are remeshed when deemed necessary.

In this chapter we review some of the techniques applied to particle methods for the solution of the diffusion equation. We compare the results of these schemes as they are applied to some simple one dimensional problems for the convection-diffusion equation. Finally we describe some recent developments of the method of particle strength exchange for the treatment of viscous boundary conditions for simple geometries and we propose a novel scheme to this effect as well.

4.1 The 1-D Convection-Diffusion Equation Model

We examine the techniques that account for diffusion in the context of particle methods (particle strength exchange, random walk, core expansion) by applying them to simple one dimensional problems governed by the following convection diffusion equation for the 'vorticity' ω :

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} = \nu \frac{\partial^2 \omega}{\partial x^2} \tag{4.1}$$

where ν is the diffusion coefficient and $u(x,t)$ is a given velocity field. In a Lagrangian formulation we follow the characteristics of the equation (described by the particle trajectories, x_a) by solving :

$$\frac{\mathrm{d}x_a}{\mathrm{d}t} = u(x_a,t) \tag{4.2}$$

Then on those characteristics Eq. 4.1 reduces to

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \nu \frac{\mathrm{d}^2 \omega}{\mathrm{d}x^2} \tag{4.3}$$

Particle methods approximate the vorticity field by a linear superposition of the vorticity field of N particles as:

$$\omega(x,t) = \sum_{i}^{N} \Gamma_i(t)\eta_\epsilon(x - x_i(t)) \tag{4.4}$$

where $\Gamma_i(t)$ is the circulation assigned to each particle and $\eta_\epsilon(x - x_i(t))$ is a function describing the vortex field, per unit circulation, induced at x by a particle located at x_i. The characteristics of the equations are determined by following the trajectories of the particles and the diffusion equation is solved on the particle locations.

It is interesting to observe here that in order to solve the diffusion equation the various particle methods apply on the different parameters of the expression of the vorticity as given in Eq. 4.4. For example, the random walk applies on the particle locations, $x_i(t)$, the particle strength exchange on the particle assigned circulation, $\Gamma_i(t)$, the core expansion technique on ϵ and another deterministic scheme (Fishelov, 1990) on η, to mention the most notable ones.

4.1.1 Test Cases

We examine the method of random walk, core expansion and particle strength exchange as they are applied to the solution of some one-dimensional problems for which exact (ω^{ex}) solutions exist. These problems are described by their initial and boundary conditions for ω and the imposed velocity field u. For all cases the viscosity (ν) of the flow is taken to be equal to 10^{-3}. Solutions are obtained for $0 \leq x \leq \infty$ for the bounded cases and for $-\infty \leq x \leq \infty$ for the unbounded ones.

Problem 1 : The initial condition is :

$$\omega(x,0) \;=\; f(x) \;=\; xe^{-x^2}$$

The boundary Conditions $\omega(0,t) \;=\; 0$ and a homogeneous velocity field $u(x,t) \;=\; 0$ define the problem.

Problem 2 : (Forced Collision of two Gaussian layers (Kambe, 1983)). The initial conditions is:

$$\omega(x,0) \;=\; f(x) \;=\; e^{-(x+2)^2} \;-\; e^{-(x-2)^2}$$

The flow is considered to be unbounded with a forcing velocity field $u(x,t) \;=\; -\alpha\, x$ where α an arbitrary constant (here $\alpha \;=\; 1$).

Problem 3 : (Rayleigh problem of an impulsively started plate). The problem is defined by homogeneous (u,ω) initial conditions and boundary conditions :$u(0,t) = 1$

We compare the numerical solutions ω^{nu} using as an error measure: RMS $=$ $\sqrt{\frac{\sum_i^N (\omega_i^{ex} - \omega_i^{nu})^2}{N}}$ where ω is computed at the particle locations ($x_i,\ i = 1, ..., N$)

4.2 Random Walk

To outline the method of the random walk we consider its application to the solution of *Problem 1.* An integral form solution to this problem is given by :

$$\omega^*(x,t) \;=\; \int_{-\infty}^{\infty} \big(G(x,y,t) - G(x,-y,t)\big) f(y)\, dy \qquad (4.5)$$

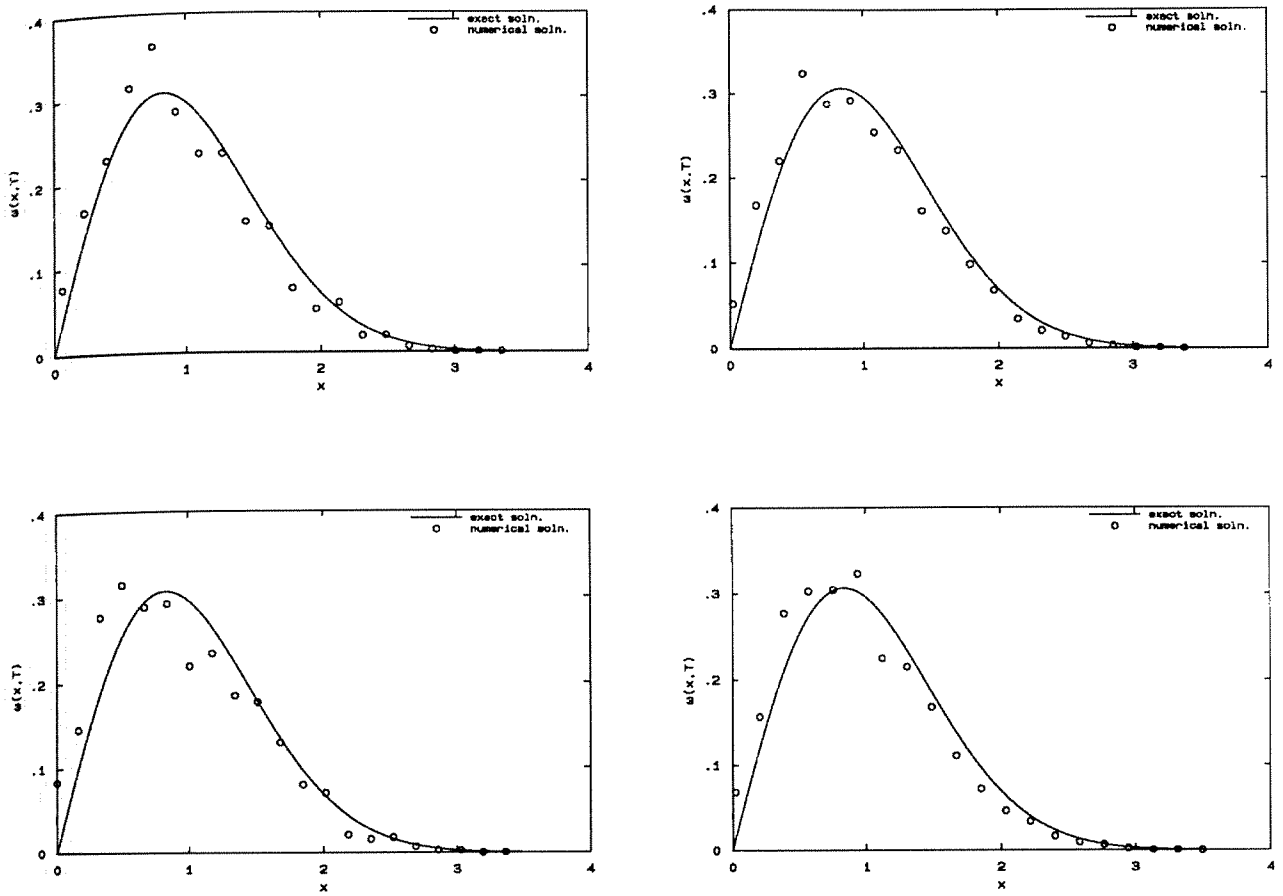


FIG. 4.1 Random walk solution of *Problem 1.* Number of particles used is (clockwise starting from the bottom left) 500,1000,2500 and 5000 respectively

where

$$G(x,y,t) = \frac{1}{\sqrt{4\pi\nu t}} e^{-(x-y)^2/4\nu t}$$

The above integral may be calculated explicitly to give:

$$\omega^{\mathrm{ex}}(x,t) \;=\; x\, e^{-x^2/(1+4\nu t)}/(1+4\nu t)^{3/2}$$

In the context of the random walk the integral solution may be interpreted probabilistically as follows: Place N randomly spaced particles on the line $x \geq 0$, in the computational domain and at positions $x_i^0, i = 1, ..., N$ and assign to each particle a strength of $f(x_i)/N$. Let then the particles undergo a random walk by changing the positions of the particles at each time step δt under the following rule :

$$x_i^{n+1} \;=\; x_i^n + \xi_i^n$$

where ξ_i^n are Gaussian independent random variables having mean 0 and variance $2\nu\delta t$. Now as we let the number N of particles go to infinity we may observe that the expected distribution of the particle strength on the x-axis would approximate Eq. 4.5

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{M} f(x_i)\ (x \le x_i \le x+dx,\ t=T,\ \ ;M \ge 0)}{Ndx} = \omega^{\mathrm{ex}}(x,t) \qquad (4.6)$$

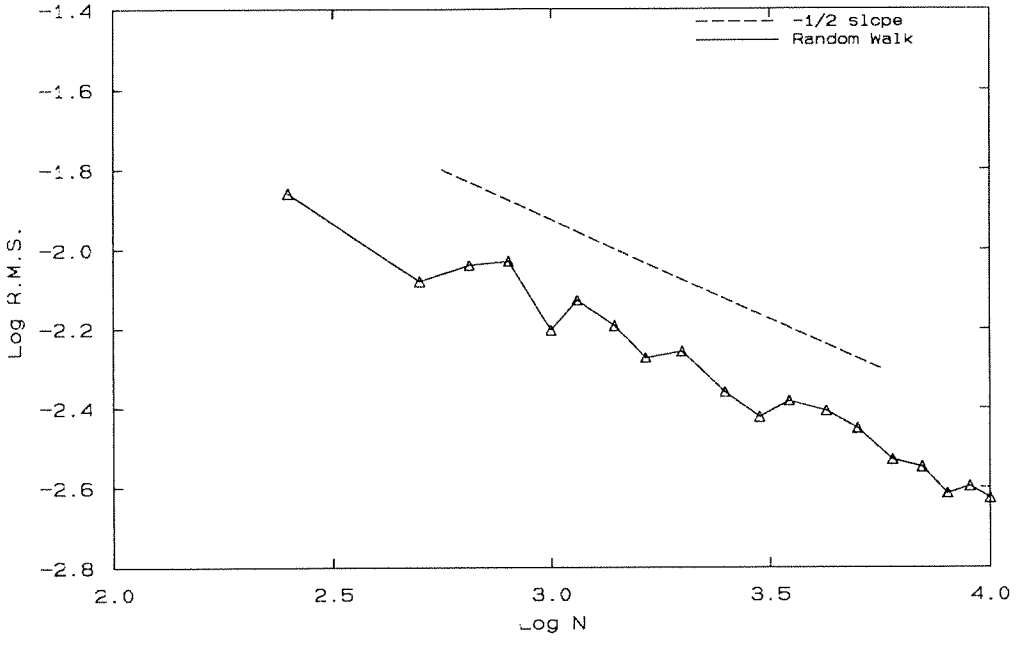


FIG. 4.2 Random walk solution of *Problem 1.* Sampling every 20/N points. $\nu = 10^{-4}$, $\delta t = .1$, $T = 10$.

In order to satisfy the homogeneous boundary condition for the vorticity we assume an odd extension of the function $f(x)$ on the line $x \le 0$ and we perform the random walk on the whole line. We may mention here that the homogeneous boundary condition could have been satisfied by the generation of particles at the boundary as is a common practice for the two-dimensional application of the method on more complicated domains. However this doesn't seem to affect the purpose of these experiments which is the evaluation of the general method. In Fig. 4.1 the results of the computation for several numbers of particles, using a time step $\delta t = 0.5$ and a sampling interval of $dx = .175$, are shown. One may observe the slow convergence of

the method which has been estimated to be of $\mathcal{O}(1/\sqrt{N})$ by Milizzano and Saffman (1977). In order to check this estimate a series of numerical experiments was performed for several numbers of particles and the *rms* error of these experiments is plotted versus the number of particles in Fig. 4.2 The characteristic law of $-1/2$ is observed.

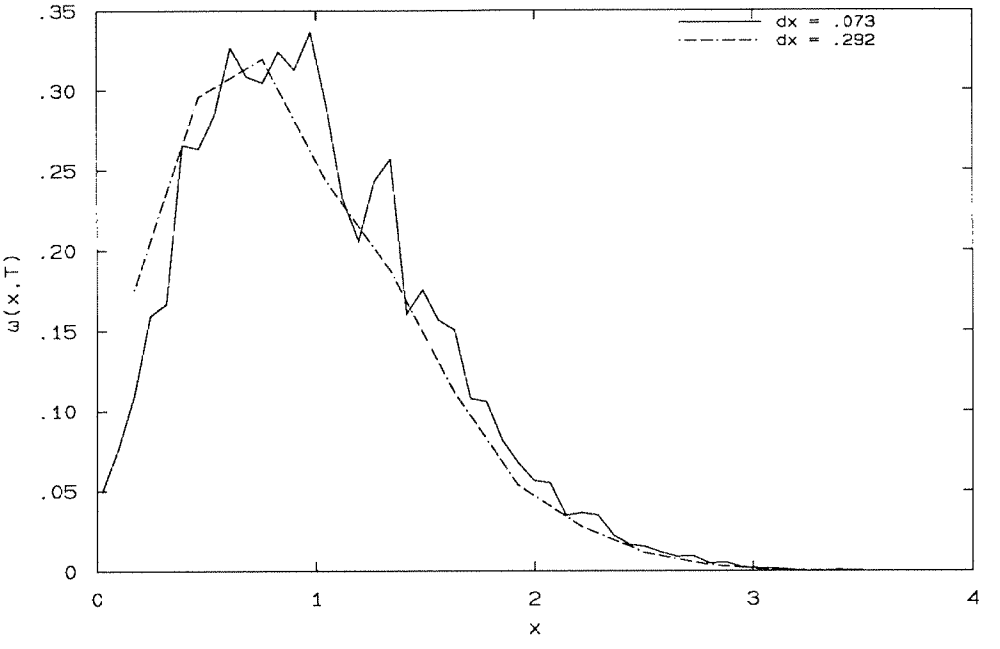


FIG. 4.3 Random walk solution of *Problem 1*. Variation of results due to sampling interval change

It is evident from these results that to get a reasonable accuracy large numbers of particles are necessary. Moreover as the viscosity of the flow increases the solution deteriorates further as the variance of the random variables becomes larger, resulting in less accurate solutions for high viscosity flows. The scheme of random walk is usualy implemented in a fractional step algorithm for the solution of the convection-diffusion equation. The particles are advanced so as to resolve the Euler equations and subsequently undergo a random walk to account for the viscous effects. In addition to the problems the method is facing in merely solving the diffusion equation, when used in a viscous splitting algorithm, it imposes a strict time step for the convection equation so that the random walk solution is not dominated by the inaccuracies in the solution of the convection step, especially when solving flows of vanishing

viscosity. This fact is common to all numerical methods that treat convection - diffusion problems and the random walk should not be considered as 'numerical diffusion' free when it is considered in that context.

We may mention here that the way one performs the sampling of the particles affects the presentation of the results. In Fig. 4.3 we plot the results for sampling increments of $dx = .073$ and $dx = .292$ for 5000 particles spread over $0 \leq x \leq 3.5$ It is evident from the computational results shown here that although the method of the random walk is an attractive technique for the solution of the diffusion equation, because of its simplicity, its slow convergence make it a questionable alternative for computations using a reasonable number of particles.

4.3 Core Expansion

The core expansion method is based (Leonard, 1980) on the linearity of the diffusion equation. As expressed by Eq. 4.4 the vorticity field may be represented by a linear superposition of elementary Gaussian vorticity fields. When these individual fields satisfy the diffusion equation so does the total vorticity field that they represent. The diffusion of each particle is achieved by assuming each one of them to be a model of an Oseen's vortex so that the core radius (ϵ) at time (t) is defined as :

$$\epsilon^2(t) = \epsilon^2(0) + 4\nu t \tag{4.7}$$

This guarantees an accurate solution of the diffusion equation (the only error introduced being the approximation of the initial condition by the particle method). However the continuous expansion of the vortex cores hurts the convection step as the error introduced by the particle approximation is of $\mathcal{O}(\epsilon^2)$ Moreover it has been shown by Greengard (1984) that this method does not lead to a consistent solution of the Navier-Stokes equations. A plausible alternative may be the remeshing of the vortex field after each diffusion step. Such an approach would fall into the category of hybrid methods and would introduce additional numerical diffusion due to the remeshing process. In the past, the method has been used with some success by Meiburg (1989) for the simulation of the two-dimensional fingering instability.

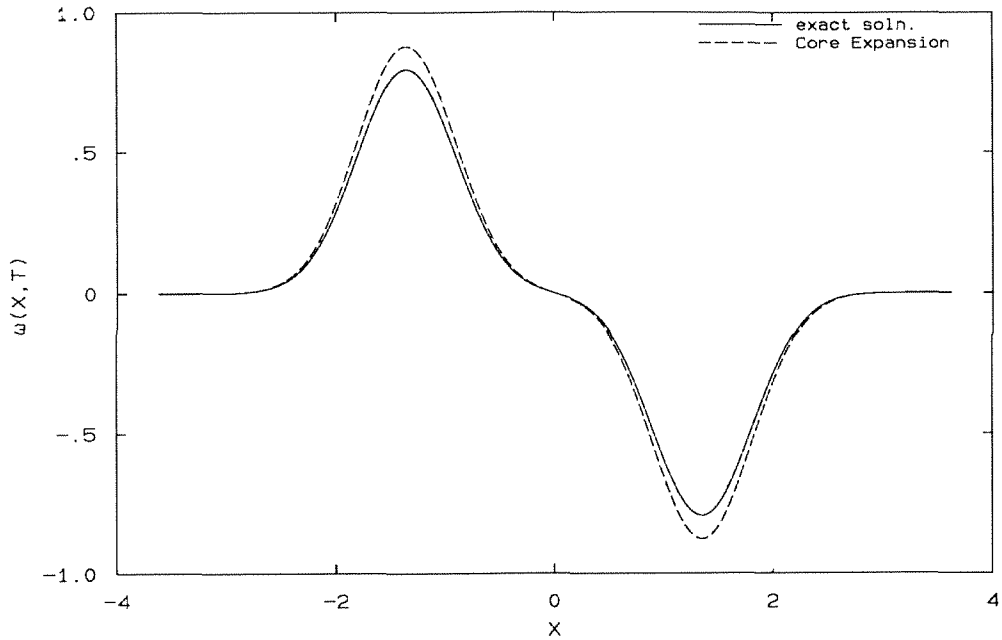


FIG. 4.4 Forced collision of Gaussian shear layers. ($\nu = 10^{-3}$, T=1)

To examine the validity of the method we use it to simulate the forced collision of two opposite signed Gaussian shear layers (*Problem 2*). In Fig. 4.4 the results of the computation are shown. We observe that the vortex calculations develop slower in time (note here that a similar observation was made by Meiburg in his simulations). Further computations have shown that by increasing the resolution the method does not reproduce the exact solution. The computational results presented here complement the argument that the method does not converge to solutions of the Navier-Stokes equations The method provides an inexpensive model for diffusion effects (especially for flows of vanishing viscosity) but it should not be used (at least without a redistribution step) if consistent solutions of the Navier-Stokes equations are sought.

4.4 Particle Strength Exchange (PSE)

In this method the diffusion equation is satisfied by modifying the strength of the particles used to describe the flow. If the particles occupy regular positions on a grid the method may be shown to be equivalent to a second order finite difference approximation (H.Cottet, 1990). In general the method simulates accurately the diffusion process provided a minimum particle overlap is maintained. For most flows of interest however the convection distorts the particle locations, due to the accumulated strain of the fluid elements, and the scheme needs to be complemented by a particle redistribution algorithm to produce accurate results.

We examine the method in more detail and we consider first its application to the solution of the diffusion equation on an unbounded domain and then its application in solving the diffusion equation with boundary conditions for simple one dimensional geometries.

4.1.1.1 Infinite Domain

The method for the whole space has been introduced by Raviart (1987) and his coworkers (Cottet (1987), Huberson (1987), Mas Gallic (1987)). It is based on the idea of approximating the Laplace operator Δ with an integral operator Δ^ϵ and applying a quadrature rule to this integral using as quadrature points the locations of the particles.

More specifically we define:

$$\Delta^\epsilon \omega \;=\; \frac{1}{\epsilon^2}(\eta_\epsilon * \omega - \omega) \tag{4.8}$$

where $\eta_\epsilon(\mathbf{x}) \;=\; \frac{1}{\epsilon^2}\eta(\mathbf{x}/\epsilon)$, is a regularization (smoothing) function. If $\eta(\mathbf{x})$ (with $\mathbf{x} = (x_1, x_2)$) is chosen so that there is an integer $m \geq 2$ such that:

$$\int_{\mathbf{R}^2} x_i^2\, \eta(\mathbf{x})\, d\mathbf{x} \;=\; 2, \quad \int_{\mathbf{R}^2} \mathbf{x}^\alpha\, \eta(\mathbf{x})\, d\mathbf{x} \;=\; 0, \quad 1 \leq |\alpha| \leq m+1$$

then it can be shown that :

$$\Delta^\epsilon \omega \;=\; \Delta\omega + \mathcal{O}(\epsilon^m) \tag{4.9}$$

The diffusion equation is replaced by the following integro-differential equation for the vorticity

$$\frac{\partial \omega}{\partial t} = \frac{\nu}{\epsilon^2} \int_{\mathbf{R}^2} [\omega(\mathbf{y}) - \omega(\mathbf{x})]\, \eta_\epsilon(\mathbf{x} - \mathbf{y})\, d\mathbf{y} \tag{4.10}$$

The integral operator is discretized using as quadrature points the locations of the particles. This discretization introduces an additional error of $\mathcal{O}(h^k/\epsilon^{k+1})$ where h is a representative inter-particle spacing and k is dependent on the properties of the function η. So finally, employing the expression Eq. 4.4 for ω, the solution of the diffusion equation is reduced to the equivalent problem of solving a set of coupled ordinary differential equations for the strengths $(\Gamma_i(t))$ of the particles. The method has been shown to be stable (Degond & Mas-Gallic, 1988) provided that there is a positive constant C_0 such that

$$\nu \leq C_0\, \epsilon^2$$

The total error introduced by the approximation of the Laplacian and the discretization on the particles is

$$\mathcal{O}(\epsilon^m + h^k/\epsilon^{k+1}).$$

The method may be expressed algorithmically in 1-D or 2-D formulation as :

$$\frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} = \frac{\nu h^d}{\epsilon^2} \sum_{j=1}^{N} (\Gamma_j^{n-1} - \Gamma_i^{n-1}) \eta_\epsilon(\mathbf{x}_i - \mathbf{x}_j), \quad d = 1, 2 \tag{4.11}$$

Pépin (1990) observed that when one uses a Gaussian regularization function and an Euler integration in time with time step

$$\delta t = \frac{\epsilon^2}{4\nu}$$

the results of the computations are independent of the number of particles one uses to resolve the equation provided they overlap. In this case Eq. 4.11 becomes :

$$\Gamma_i^n = \Gamma_i^{n-1} + \left(\frac{h}{\sqrt{4\pi\nu\delta t}}\right)^d \sum_{j=1}^{N} (\Gamma_j^{n-1} - \Gamma_i^{n-1}) e^{-(\mathbf{x}_i - \mathbf{x}_j)^2/4\nu\delta t}, \quad d = 1, 2 \tag{4.12}$$

The above result may be explained noting that the exact solution of the diffusion equation for an unbounded (1-D) domain is expressed as :

$$\omega(x, t + \delta t) = \int \frac{exp(-(x-y)^2/\sqrt{4\nu\delta t})}{4\pi\nu\delta t}\, \omega(y, t)\, dy \tag{4.13}$$

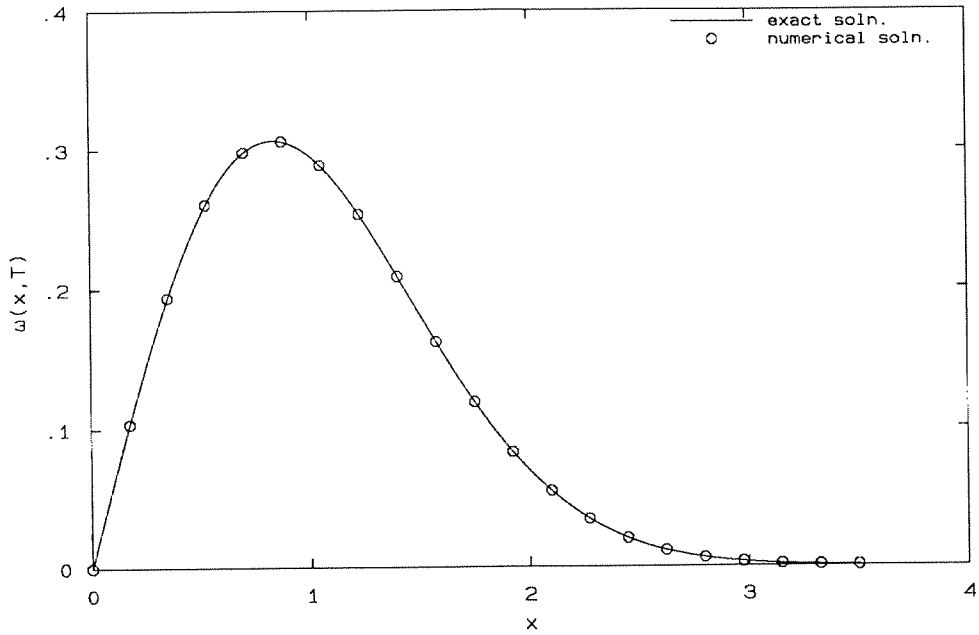


FIG. 4.5 Particle Strength Exchange solution of *Problem 1.* $T = 100,\ , \delta t = .5,\ N = 80$

The kernel is such that total circulation is conserved so we may write

$$\omega(x, t+\delta t) \ - \ \omega(x,t) \ = \ \int \frac{exp(-(x-y)^2/4\nu\delta t)}{\sqrt{4\pi\nu\delta t}} \left(\omega(y,t) \ - \ \omega(x,t)\right) dy \qquad (4.14)$$

Setting the strength of the particles to be equal to

$$\Gamma_i \ = \ \omega(x_i)\, h^d, \qquad \mathrm{d} \ = \ 1, 2$$

and discretizing Eq. 4.14 using as quadrature points the locations of the particles results in Eq. 4.12. This justifies the existence of an *optimum step* for the method of particle strength exchange as then the only error is that of the quadrature rule that is known to vanish provided the particles overlap.

N	250	375	500	750	1000
Log R.M.S.	-5.24204	-5.24343	-5.24579	-5.24666	-5.24689

TABLE 2.2.1 Convergence study for $h/\epsilon = 1$.

This theoretical estimate is indeed verified by the results shown in Table 1 for the solution of *Problem 1* using various numbers of particles. The error remains constant for large enough N. It is not exactly zero as during the diffusion process we used fixed locations of particles and we did not add particles to pick up the vorticity that has diffused to the outer parts of the domain at time T. If one wishes not to operate on the optimum overlap ratio then the behavior of the error is shown in Fig.4.6 for different values of ϵ. As we can observe the error decreases rapidly as we decrease the overlap ratio. When for each ϵ the number of particles N, is such that the overlap ratio becomes equal or less than one then the error curve reaches a plateau . We have reached the minimum error and increasing the number of particles doesn't improve our solution. The results shown on Fig.4.6 may be compared with those of Fig. 4.1 for the random walk solution of *Problem 1*. One can observe then that for the same number of particles and the same time step the results of the P.S.E. method are more than two orders of magnitude superior to those obtained using the random walk.

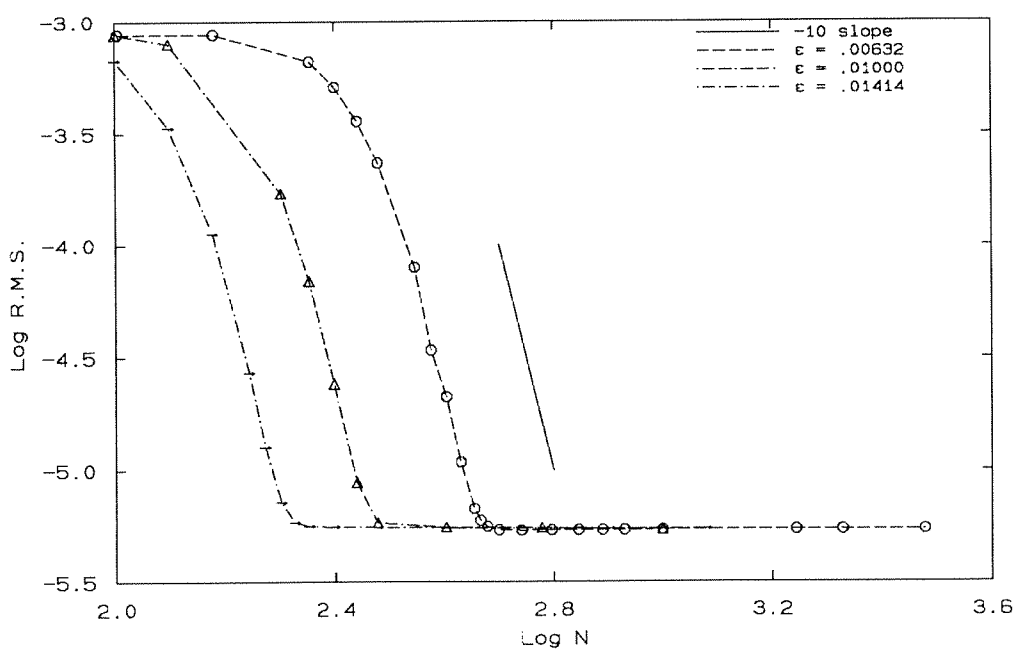


FIG. 4.6 Particle Strength Exchange solution of *Problem 1*. $\nu = 10^{-4}$, $\delta t = .1$, $T = 10$

Note also that the method does not require a viscous splitting algorithm for the solution of the convection-diffusion equation. Eq. 4.2 and Eq. 4.11 may be integrated

simultaneously to account for convection and diffusion by modifying the particle location and strength respectively. This observation helps reduce the computational cost of such methods as the smoothing kernel ($\eta_\epsilon(x)$) needs to be calculated only once at the particle locations.

4.5 What if Particles are not Uniformly Distributed ?

The accuracy of the scheme of particle strength exchange is highly dependent on the particle overlap as theoretical analysis and practical results dictate. Such a requiremnt is necessary of the convection step as well when one uses smooth vortices for the solution of the Euler equations.

The distribution of the computational elements is strongly dependent on the flow map. The local strain may generate a substantial clustering of particles in one direction accompanied by an expansion in another direction. An initially uniform distribution of overlapping particles may become totally distorted as a flow map rearranges the particle positions. When overlap is lost information cannot be transferred between the particles and the accuracy of the simulation deteriorates.

These facts are demonstrated by the following simple one-dimensional computational experiments. At first we repeat the simulation of *Problem 2*. For this case the particles overlap at all times and the overlap ($r = \delta x/\delta t$) remains uniform throughout the computational field ($r(t) = r(0)exp(-2at)$) at all times.

The solution obtained from the particles is indistinguishable from the exact so in Fig.4.7 we show the error of the simulation as a function of time. Note that the error decreases with the progress of time as the overlap is increased. This case is optimally solved by the particle method as the uniformity of the particle distribution is maintained throughout the course of the computation.

The simulation is repeated now but this time a velocity field of the form

$$u(x) = a\frac{sin(5\pi x)}{1+x^6}$$

is used to convect the particles. Due to the time independent stagnation points induced by this velocity field particles accumulate around these points, requiring a smaller time step for the integration of the equations, whereas (if no local re-griding takes place) other regions of the domain are devoid of particles.

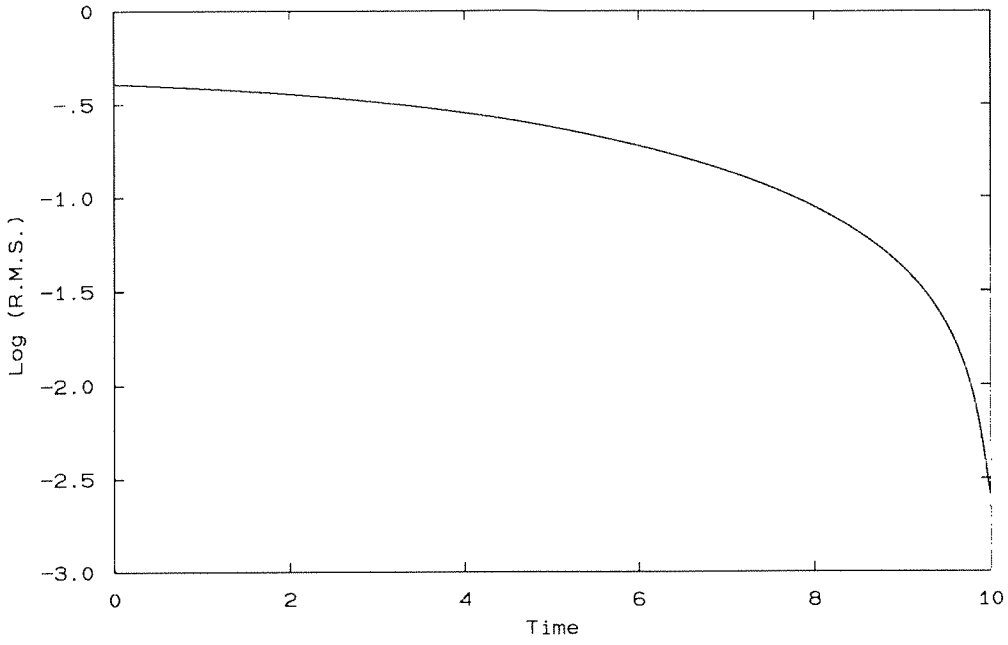


FIG. 4.7 Collision of two Gaussians. Particle Strength Exchange Solution error

Since no analytic solution exists for this problem we compare the results of the particle method (using the PSE scheme for the diffusion step) with the results obtained from the careful integration of a finite difference scheme using the method of lines with cubic Hermite interpolation.

In Fig. 4.8, Fig. 4.9 we show the results of the computation and we monitor the particle overlap for different times. Note that although initially at certain regions the particles do not overlap the agreement is excellent. However the situation deteriorates with time and later unphysical oscillations are observed in the solution due to the loss of particle overlap. A remedy could be a regridding procedure that restores a uniform particle distribution when deemed necessary conserving the properties of the field. Such a procedure is described in Ch.6 for the application of the method in a two-dimensional domain.

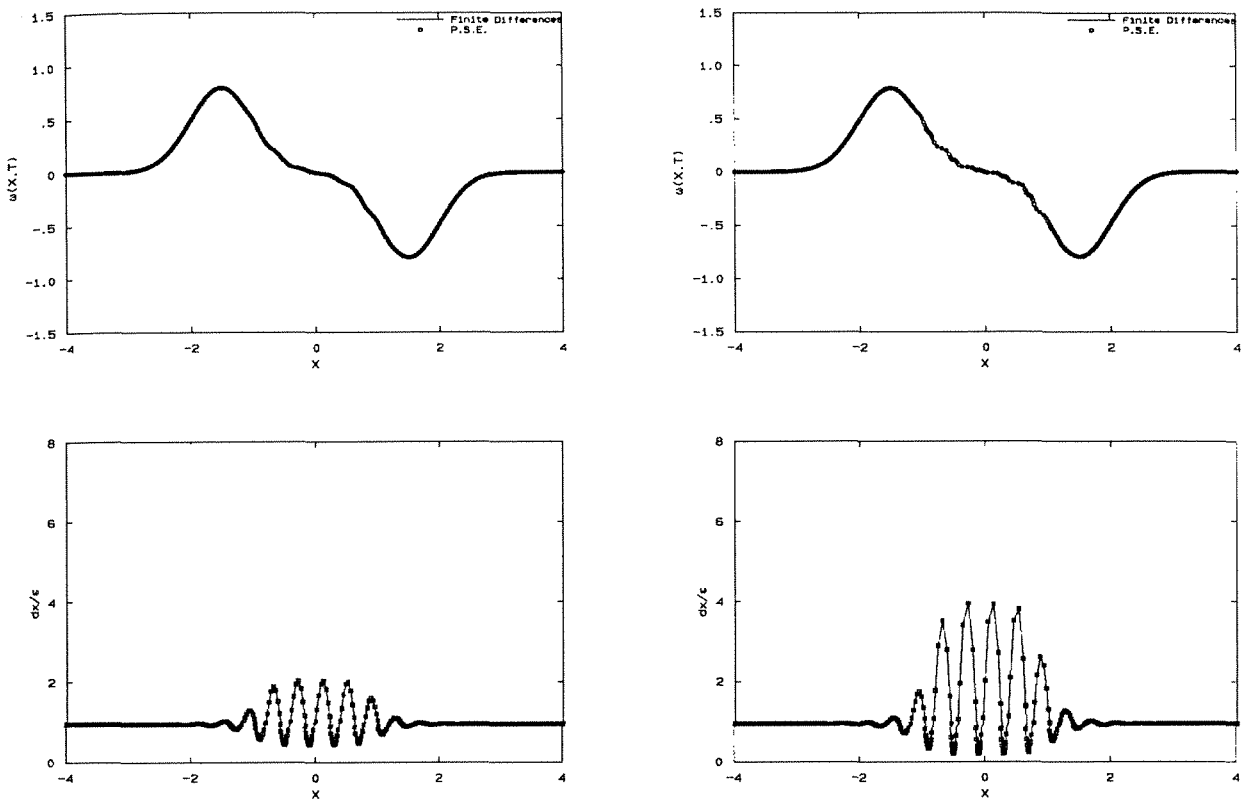


FIG. 4.8 Forced Interaction of Two Gaussians. Vorticity values (top) and respective overlap at T=.1 (left) and T=.20. (Solid line : finite differences, Symbols : Scheme of PSE)

4.6 Boundary Conditions for 1-D Convection - Diffusion Equations

As we have used the one-dimensional world to test numerical schemes for the representation of diffusion it seems appropriate to formulate the problem of enforcing boundary conditions as well, in 1-D domains. The task of enforcing boundary conditions in the context of a particle method is not a trivial one. The difficulty is twofold. In the Lagrangian world the boundary conditions may be seen as a process where information (values of the field variable) needs to be transferred from the boundary to the particles. This requires that particles resolve the region near the boundary. However the irregularity of the particle distribution makes the process cumbersome and difficult to analyze. Moreover for physically interesting problems the boundary conditions are in terms of variables other than those that are resolved on the particles (e.g., the no-slip boundary condition for vortex methods). The boundary conditions (e.g., on the velocity) need to be transformed to equivalent ones (e.g., on the vorticity) to be implemented in a particle scheme.

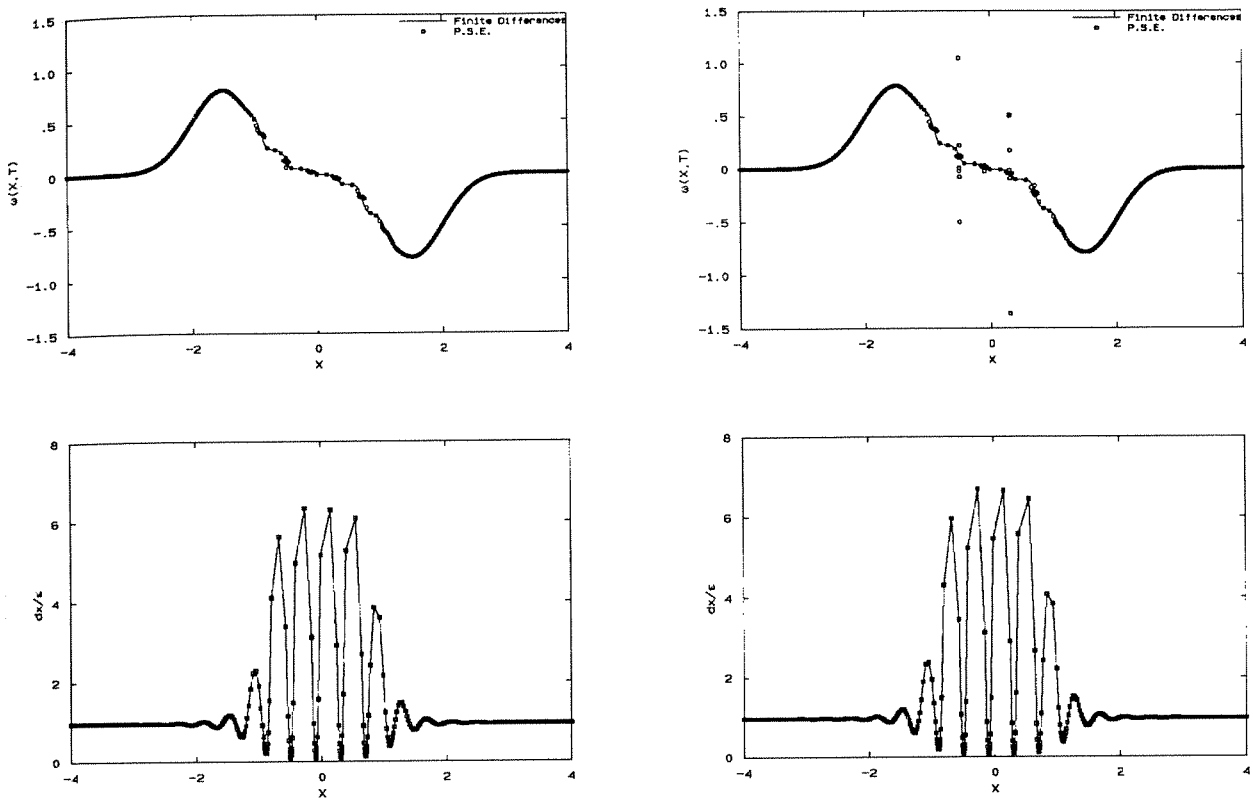


FIG. 4.9 Forced Interaction of Two Gaussians. Vorticity values (top) and respective overlap at T=.3 (left) and T=.314.(Solid line : finite differences, Symbols : Scheme of PSE)

The issue has not been satisfactorilly resolved in the past and research is active in this area. Here we describe some recent efforts in deriving boundary conditions for some simple one dimensional geometries. In addition we propose a new scheme that accurately resolves boundary conditions of the Neumman and Dirichlet type for simple geometries. The extension of this scheme to higher dimensions is discussed in Ch.5

Method I : The Particle Strength Exchange method has been recently extended to cover boundary conditions for simple 1-D geometries by S. Mas-Gallic (1990) and B.Luquin and S.Mas-Gallic (1990) in a conservative formulation.

For the sake of completeness we repeat here the basic ideas of the derivation of the method : Let ω be the vorticity field in a domain Ω with boundary $\partial\Omega$ and define the operator $\mathcal{L} = I - \epsilon^2 \Delta$ and f such that:

$$\mathcal{L}\omega = \omega - \epsilon^2 \Delta\omega = f, \ in\, \Omega$$

We convolve the above equation with G_ϵ where G_ϵ is the Green's function for the infinite domain of the operator $\mathcal{L}$.

$$\omega * G_\epsilon - \epsilon^2 \Delta\omega * G_\epsilon = f * G_\epsilon \tag{4.15}$$

Using now Green's identities on the left hand side of the equation we obtain :

$$\omega * G_\epsilon - \epsilon^2 \omega * \Delta G_\epsilon + \epsilon^2 \int_{\partial\Omega} \omega \frac{\partial G_\epsilon}{\partial n} ds - \epsilon^2 \int_{\partial\Omega} G_\epsilon \frac{\partial \omega}{\partial n} ds = f * G_\epsilon \tag{4.16}$$

By definition $G_\epsilon - \epsilon^2 \Delta G_\epsilon = \delta$, so that :

$$\omega = -\epsilon^2 \int_{\partial\Omega} \omega \frac{\partial G_\epsilon}{\partial n} ds + \epsilon^2 \int_{\partial\Omega} G_\epsilon \frac{\partial \omega}{\partial n} ds + f * G_\epsilon$$

However solving Eq. 4.15 for $\Delta\omega$ and assuming $f \approx \omega$ we obtain the following integral approximation for the Laplacian.

$$\Delta^h \omega = \frac{1}{\epsilon^2}(\omega * G_\epsilon - \omega) - \int_{\partial\Omega} \omega \frac{\partial G_\epsilon}{\partial n} ds + \int_{\partial\Omega} G_\epsilon \frac{\partial \omega}{\partial n} ds \tag{4.17a}$$

Now this equation may be complemented by the equation that gives the values of ω on the boundary $\partial\Omega$

$$\frac{1}{2}\omega = \frac{1}{\epsilon^2}(\omega * G_\epsilon - \omega) - \int_{\partial\Omega} \omega \frac{\partial G_\epsilon}{\partial n} ds + \int_{\partial\Omega} G_\epsilon \frac{\partial \omega}{\partial n} ds \tag{4.17b}$$

to give us the full set of equations to solve. Now given the vorticity (ω) or the flux of vorticity ($\partial\omega/\partial n$) on the boundary we solve Eq.4.17b for ($\partial\omega/\partial n$) or ω respectively and substitute in Eq.4.17a to obtain the approximation for the Laplacian operator when boundaries are present. The function G_ϵ is not known a priori for all the domains Ω with arbitrary boundaries $\partial\Omega$. However it can be approximated by functions η in Ω and ζ on $\partial\Omega$ where the functions η, ζ are related appropriately in order for the method to converge and to be stable.

For example for the solution of the heat equation on the half space $\mathbf{R}^2_+$ the set of equations may be expressed as (Mas-Gallic, 1990) :

$$\begin{aligned} \Delta^h \omega(\mathbf{x}) = & \frac{1}{\epsilon^2} \int_{\mathbf{R}^2_+} (\omega(\mathbf{y}) - \omega(\mathbf{x}))\, \eta_\epsilon(\mathbf{y}-\mathbf{x})\, d\mathbf{y} \\ & - \int_{\mathbf{R}} \omega(\mathbf{x}-\mathbf{y}) \frac{\partial \zeta_\epsilon(\mathbf{y})}{\partial \mathbf{y}} d\mathbf{y} + \int_{\mathbf{R}} \zeta_\epsilon(\mathbf{x}-\mathbf{y}) \frac{\partial \omega}{\partial \mathbf{y}} d\mathbf{y} \end{aligned} \tag{4.18a}$$

for $\mathbf{x} = (x1, x2)$ in $\mathbf{R}^2_+$ and

$$\begin{aligned}\frac{1}{2}\omega(\mathbf{x}) &= \frac{1}{\epsilon^2}\int_{\mathbf{R}^2_+}\left(\omega(\mathbf{y})-\omega(\mathbf{x})\right)\zeta_\epsilon(\mathbf{y}-\mathbf{x})\,d\mathbf{y} \\ &\quad -\int_{\mathbf{R}}\omega(\mathbf{x}-\mathbf{y})\frac{\partial\zeta_\epsilon(\mathbf{y})}{\partial\mathbf{y}}\,d\mathbf{y} + \int_{\mathbf{R}}\zeta_\epsilon(\mathbf{x}-\mathbf{y})\frac{\partial\omega(\mathbf{y})}{\partial\mathbf{y}}\,d\mathbf{y}\end{aligned} \tag{4.18b}$$

for $\mathbf{x} = (x1, 0)$ on $\mathbf{R}$ where η obeys Eq. 4.8 and is related to ζ by

$$\int_{\mathbf{R}^2_+} y_2\eta(y_1, y_2+s)\,ds = \frac{\int_{\mathbf{R}^2_+} z_2\,\eta(z)\,dz}{\int_{\mathbf{R}}\zeta(z_1,0)\,dz_1}\int_{\mathbf{R}}\zeta(z,s)\,dz \qquad \text{for all } s \geq 0$$

This method has been shown by S.Mas-Gallic (1990) and B.Luquin and S.Mas-Gallic (1990) to converge to solutions of the diffusion equation with boundary conditions in the Dirichlet or Neumann form.

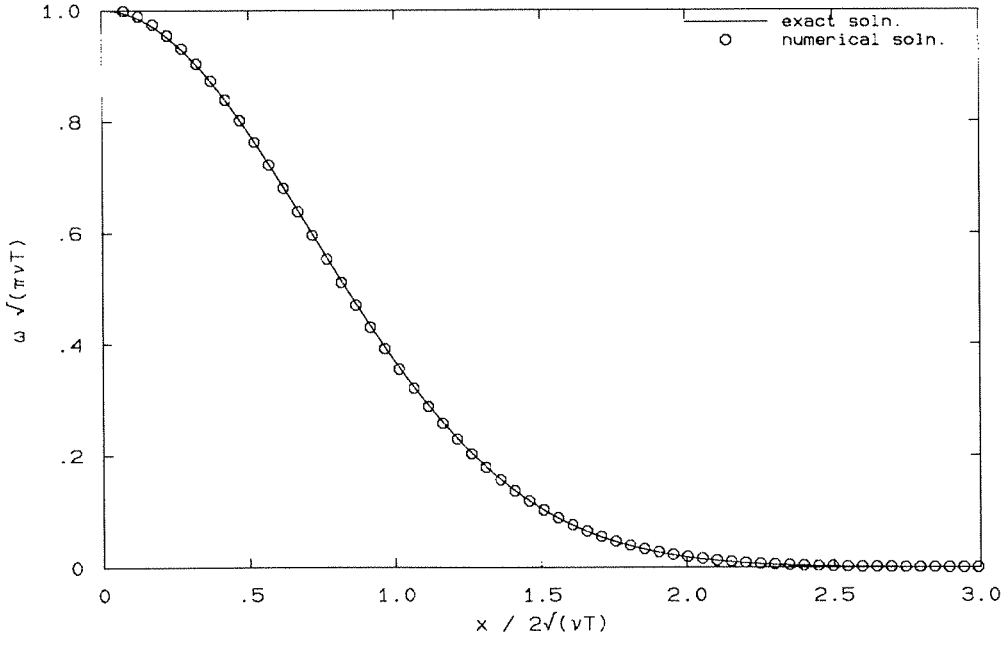


FIG. 4.10 Particle Strength Exchange solution of *Problem 3.* ($\nu = 10^{-4}$, $T = 1.$, $N = 100$)

Returning now to the 1-D space we may apply this method to the solution of the test problems already posed. If we pick $\zeta_\epsilon(x) = \eta_\epsilon(x) = \theta_\epsilon(x) = e^{-|x|/\epsilon}/2\epsilon$ then the Laplace operator may be expressed as

$$\Delta^h\omega(x) = \frac{1}{\epsilon^2}\int_0^\infty \left(\omega(y)-\omega(x)\right)\left(\theta_\epsilon(x-y)+\theta_\epsilon(x+y)\right)dy - 2\frac{\partial\omega(0,t)}{\partial x}\,\theta_\epsilon(x) \quad (4.19a)$$

for the Neumann boundary conditions on the vorticity or

$$\Delta^h\omega(x) = \frac{1}{\epsilon^2}\int_0^\infty \left(\omega(y)-\omega(x)\right)\left(\theta_\epsilon(x-y) - \theta_\epsilon(x+y)\right)dy + \omega(0,t)\frac{\theta_\epsilon(x)}{\epsilon} \quad (4.19b)$$

for the Dirichlet boundary condition on the vorticity

In Fig. 4.10 the solution of the Rayleigh problem (*Problem 3*) is shown using the Neumann boundary condition after the steady state has been reached. One may observe the good agreement of the method with the analytic solution using a small number of particles.

Method II : An alternative technique has been proposed in the context of the deterministic particle strength exchange for the treatment of the boundary conditions by Pépin (1990). We examine this method and analyze its accuracy and consistency by repeating at first the derivation of the method for the solution of the 1-D diffusion equation.

Convolve the diffusion equation with the smoothing function η_ϵ in order to get :

$$\int_0^\infty \frac{\partial\omega}{\partial t}(y)\,\eta_\epsilon(x-y)\,dy = \nu\int_0^\infty \frac{\partial^2\omega}{\partial x^2}(y)\eta_\epsilon(x-y)\,dy$$

Applying now Green's identity and assuming that the time derivative of the smoothed vorticity field is approximately equal to the time derivative of the real vorticity field results in the following equation :

$$\frac{\partial\omega(x)}{\partial t} = -\nu\,\frac{\partial\omega}{\partial x}(0)\,\eta_\epsilon(x) + \frac{\nu}{\epsilon^2}\int_0^\infty \frac{\partial\omega}{\partial x}(x+z)\,z\,\eta_\epsilon(|z|)\,dz \quad (4.20)$$

where $z = y - x$.

Expanding $\partial\omega(x+z)/\partial x$ around x using a 2nd order scheme yields :

$$\frac{\partial\omega}{\partial x}(x+z) \approx \frac{3}{2}[\omega(x+z)-\omega(x)] + \frac{1}{2}[\omega(x-z)-\omega(x)]$$

and substituting in Eq. 4.20 we get that :

$$\begin{aligned}\frac{\partial\omega(x)}{\partial t} = -\nu\frac{\partial\omega}{\partial x}(0)\eta_\epsilon(x) + \frac{\nu}{\epsilon^2}\int_0^\infty \frac{3}{2}[\omega(x+z)-\omega(x)]\,\eta_\epsilon(|z|)\,dz \\ + \frac{\nu}{\epsilon^2}\int_{-\infty}^\infty \frac{1}{2}[\omega(x-z)-\omega(x)]\eta_\epsilon(|z|)\,dz\end{aligned} \tag{4.21}$$

Comparing now the above equation Eq. 4.21 and Eq. 4.10 that expresses the method of the particle strength exchange for the unbounded domain it seems reasonable to make the approximation:

$$\int_0^\infty [\omega(x-z)-\omega(x)]\eta_\epsilon(|z|)\,dz \approx \int_0^\infty [\omega(x+z)-\omega(x)]\eta_\epsilon(|z|)\,dz \tag{4.22}$$

and express finally the method as the following integro-differential equation for the vorticity

$$\frac{\partial\omega(x)}{\partial t} = -\nu\frac{\partial\omega}{\partial x}(0)\,\eta_\epsilon(x) + \frac{2\nu}{\epsilon^2}\int_0^\infty [\omega(y)-\omega(x)]\,\eta_\epsilon(|x-y|)\,dy \tag{4.23}$$

We recover then, the expression for the unbounded domain (at least away from the boundary) and get also an additional term to account for the boundary condition.

Note that the approximation as expressed by Eq. 4.22 is an exact relation when the integration is carried over the whole real line (i.e., the limits of integration are not $[0,\infty]$ but $[-\infty,\infty]$) as z is a dummy variable. It is also a good approximation for points (x) away from the boundary as the smoothing function decays rapidly for large values of z. However close to the boundary ($x=0$) the above approximation does not lead to a consistent approximation of the Laplacian by an integral operator, similar at least to the one given by Eq. 4.8 This observation may be alternatively expressed by the following argument. Consider as a smoothing function the Gaussian. Then in order for the method to be consistent and conserve the total circulation we need that

$$\frac{1}{\sqrt{\pi}\epsilon}\int_{-\infty}^\infty e^{-(x-y)^2/\epsilon^2}\,dy = 1$$

On the other hand for the bounded domain we get

$$\frac{1}{\sqrt{\pi}\epsilon}\int_0^{\infty} e^{-(x-y)^2/\epsilon^2}\,dy \;=\; \frac{1}{2}+\frac{1}{2}erf(x/\epsilon)$$

Note now that as $x/\epsilon \geq 3$ then $erf(x/\epsilon) \approx 1$ and hence we obtain the correct behavior of the function. However as we keep a finite ϵ and $x \to 0$ then $erf(x/\epsilon) \approx 0$ and the method is not a consistent one.

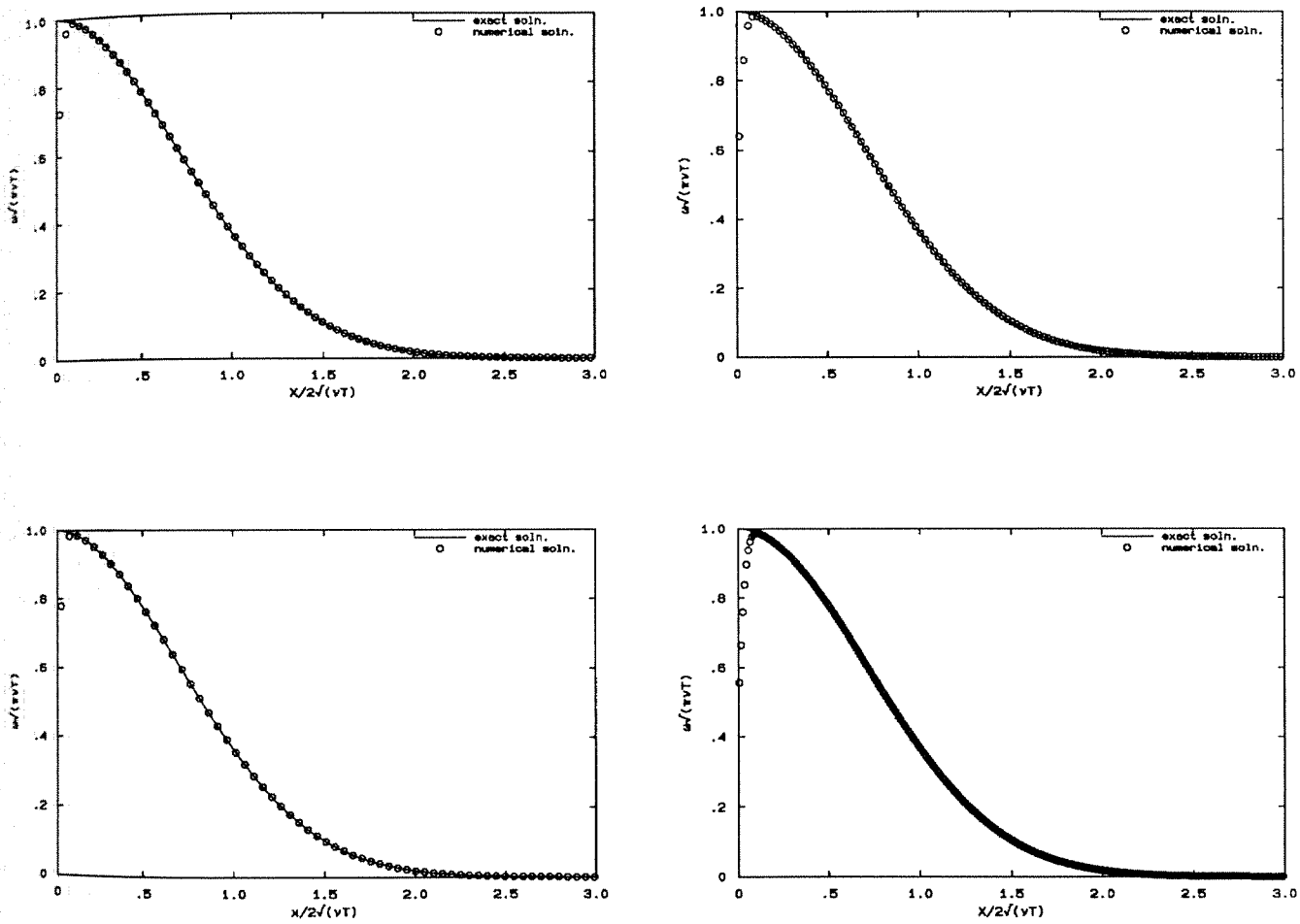


FIG. 4.11 Particle Strength Exchange solution of *Problem 3*. ($\nu = 10^{-4}$, $T = 1.$, $\epsilon = 10^{-3}$). Number of particles used (N) and respective overlap ratio (dx/ϵ) is (clockwise starting from bottom left): 60 (.99), 75 (.80), 120 (.50), 300 (.20)

In Fig.4.11 the results are shown for the solution of *Problem 3* using different numbers of particles. We observe then that for a small region ($\approx 3\epsilon$) close to the

boundary the method fails to solve accurately the diffusion equation. Increasing the number of particles reveals a trend of the method to resolve a thin 'boundary-layer' like solution. This inaccuracy seems not to affect the solution at distances farther away from the boundary and seems to be minimized when the 'optimum' time step ($\delta t = \epsilon^2/4\nu$) is employed.

Summarizing, the above method is a conservative one as can be verified by Eq. 4.23, in the sense that it gives to the fluid the correct amount of total circulation. It is consistently solving the diffusion equation at points away from the boundary. However close to the boundary the method fails to give an accurate solution of the diffusion equation. The implications of this defect cannot be readily analysed and one can only speculate about the error introduced merely by examining the accuracy of the results.

Method III : We present here an alternative approach for the solution of the diffusion equation with boundary conditions, in the context of particle methods. The method involves the solution of the diffusion equation for an even vorticity field in the unbounded domain and the use of appropriate Green's functions in order to satisfy the boundary conditions.

Consider again the diffusion equation for the vorticity with Neumann boundary conditions (a similar analysis applies for Dirichlet boundary conditions as well).

$$\frac{\partial \omega}{\partial t} = \nu \frac{\partial^2 \omega}{\partial x^2} \tag{4.24a}$$

$$\omega(x,0) = g(x) \ (x \geq 0), \quad \frac{\partial \omega}{\partial x}(0,t) = f(t) \tag{4.24b}$$

In order to solve the above equation and obtain the solution at time $t + \delta t$ after having solved for time t we decompose the problem into two sub-problems. Then at each time step we perform the following two substeps :

$$\left\{ \begin{array}{ll} 1. & \dfrac{\partial \omega_1}{\partial t} = \nu \dfrac{\partial^2 \omega_1}{\partial x^2} \\ & \omega_1(x,0) = \omega(x.t - \delta t) \\ & \dfrac{\partial \omega_1}{\partial x}(0,\delta t) = 0 \end{array} \right\} \quad \oplus \quad \left\{ \begin{array}{ll} 2. & \dfrac{\partial \omega_2}{\partial t} = \nu \dfrac{\partial^2 \omega_2}{\partial x^2} \\ & \omega_2(x,0) = 0 \\ & \dfrac{\partial \omega_2}{\partial x}(0,\delta t) = f(t) = \text{const.} \end{array} \right\}.$$

The solution at time $(t+\delta t)$ is given by the superposition of the solutions of substeps 1 and 2.

In order to solve Step 1 we have to satisfy the homogeneous Neumann boundary condition. This is achieved by extending evenly the vorticity field (ω). To solve then the diffusion equation one of the previously discussed methods for the unbounded domain may be implemented. In the current computations we used the particle strength exchange technique so that for the 1-D case we may express the solution of the problem as the solution of the following integro-differential equation:

$$\frac{\mathrm{d}\omega_1(x)}{\mathrm{d}t} = \frac{\nu}{\epsilon^2}\int_0^\infty \left(\omega_1(y)-\omega_1(x)\right)\left(\eta_\epsilon(|x-y|) + \eta_\epsilon(|x+y|)\right) dy \tag{4.25}$$

Applying again a quadrature rule to the above equation we may reduce its solution to the solution of a system of differential equations for the strength of the particles. Note that this step may be solved also by the random walk technique.

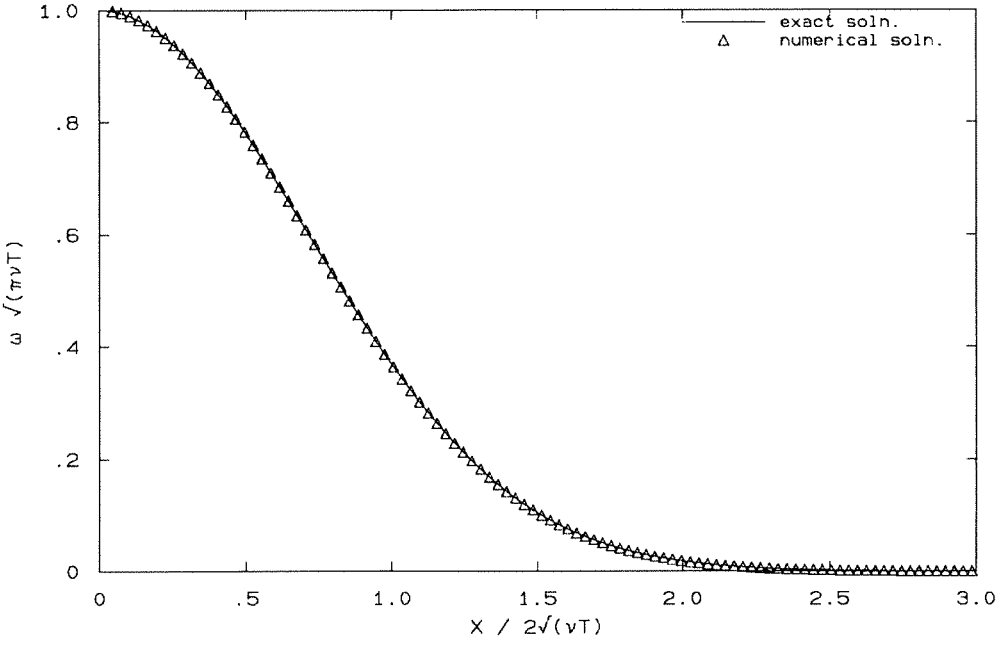


FIG. 4.12 Method III solution of *Problem 3.* ($\nu = 10^{-4}$, $T = 1.$, $N = 100$)

In order to solve Step 2 we need now to obtain the Green's function for the diffusion equation with Neumann boundary conditions. For the 1-D case there is an analytic solution to this problem expressed as :

$$\frac{\omega_2(x,\delta t)}{C} = \sqrt{\frac{4\nu\delta t}{\pi}}e^{-x^2/4\nu\delta t} - x\left(1 - erf(x/4\nu\delta t)\right)$$

The final solution of the 1-D problem is given then as:

$$\begin{aligned}\omega(x,t) &= \omega_1 + \omega_2 \\ &= \frac{\nu}{\epsilon^2}\int_0^\infty \left(\omega_1(y)-\omega_1(x)\right)\left(\eta_\epsilon(|x-y|)+\eta_\epsilon(|x+y|)\right)dy \\ &+ \sqrt{\frac{4\nu\delta t}{\pi}}e^{-x^2/4\nu\delta t} - x\left(1 - erf(x/4\nu\delta t)\right)\end{aligned}$$

For the 1-D case the convergence of this method is governed by the convergence of the particle strength exchange for the unbounded domain. In Fig.4.12 the results of the present method are shown for the solution of the Rayleigh problem using the same parameters as the results shown in Fig. 4.10. Computational and analytical results are in excellent agreement

Note however that the scheme requires the existence of particles in the immediate neighborhood of the boundary so that the effect of the boundary condition is accurately transferred to the domain. Moreover the employed even extension of the solution in the domain (method of images) makes the extension of the present scheme to higher dimensions a difficult one especially for complex geometries. In Ch.5 we discuss a viscous splitting algorithm that enables us to extend the method to higher dimensions by incorporating the idea of obtaining an analytic solution for the Neumann problem with homogeneous initial conditions.

CHAPTER 5

Boundary Conditions for Viscous Vortex Methods

Consider a two-dimensional body (whose surface is defined by the unit vectors $\mathbf{s}, \mathbf{n}$), translating with velocity $\mathbf{U_b}(t)$ and rotating with angular velocity $\Omega(t)$, in an incompressible viscous flow field induced by a uniform flow ($\mathbf{U}_\infty(t)$) and vorticity ($\omega_f(\mathbf{x}, t)$) in the wake. The boundaries of the body are the source of vorticity generation in the fluid. This physical mechanism is based on the experimental fact that the velocity ($\mathbf{u}$) of the fluid at immediate contact with the body is equal to the velocity of the body ($\mathbf{U_b}$), the **'no-slip'** boundary condition:

$$\mathbf{u}(\mathbf{x}_s) \;=\; \mathbf{U_s} \tag{5.1}$$

with

$$\mathbf{U_s} \;=\; \mathbf{U_b} \;+\; \Omega(t)\,\widehat{\mathbf{e}_z} \times (\mathbf{x}_s \;-\; \mathbf{x}_b)$$

where $\mathbf{x}_s$ denotes the position of the surface of the body and $\mathbf{x}_b$ the location of its center of mass.

Vorticity may be considered equivalent to the rotation of fluid elements implied by velocity gradients. As the velocity of the fluid far from the wall is not directly influenced by its presence, velocity gradients are established that result in vorticity generation in the fluid. The 'no-slip' boundary condition enters in the mathematical formulation of the Navier-Stokes equations usually in the form described by Eq. 5.1. However numerical schemes that rely on the vorticity formulation of the Navier-Stokes equations need an equivalent boundary condition for the vorticity field. Such a boundary condition has to be consistent with the kinematics that relate the vorticity field and the velocity field and model the physical mechanism of vorticity creation at the boundary.

5.1 Vorticity Boundary Conditions

The velocity field and the vorticity field are related by the following two *kinematic* relationships :

$$\nabla \times \mathbf{u} = \boldsymbol{\omega} \tag{5.2}$$

with :

$$\nabla \cdot \mathbf{u} = 0 \tag{5.3}$$

As the boundary conditions are in terms of the velocity the respective vorticity field (and the vorticity boundary condition) has to obey the above two kinematic restrictions.

The above equations may be replaced by an equivalent Poisson's equation for the velocity (resulting by taking the curl of Eq. 5.2 and using Eq. 5.3) :

$$\nabla^2 \mathbf{u} = -\nabla \times \boldsymbol{\omega} \tag{5.4}$$

or introducing the streamfunction (Ψ) (with $\mathbf{u} = \nabla \times \Psi \hat{\mathbf{e}_z}$)

$$\nabla^2 \Psi = -\omega \tag{5.5}$$

We may distinguish two types of numerical schemes that are used to resolve the above equations. The differential formulation (Roach (1972)) relies on the direct discretization of the above equations using a finite difference scheme. In this formulation the far field boundary condition dictates a computational domain that is larger than the rotational part of the flow. This might be viewed as contradicting the benefits of choosing the vorticity field as the primary variable in the computations but it does not pose an unsurmounted obstacle for the practitioners of the method. Alternatively one may consider the equivalent integral formulation of the above equations. In this approach the velocity field is determined using the Green's function solution of Eq. 5.4 (the Biot-Savart law). An advantage of this formulation is the implicit enforcement of the far field boundary condition making it suitable for vorticity based formulations as there is no need to solve for the non-vortical part of the domain.

The form of the appropriate boundary condition relies on the choice of formulation to determine the solution in the rest of the computational domain. To solve the equations resulting from the discretization of the differential forms, the value of the vorticity field on the boundary is necessary. One-sided difference formulas may be used to determine the boundary vorticity (see Roache (1972)). In the past first order formulas have been implemented to ensure stability and consistency of the numerical scheme. However as it is discussed and demonstrated by Wu (1976) such computations are of low accuracy and fail to account correctly for the effects of the pressure gradient on the boundary. Recently Hou and Wetton (1992) have devised finite difference formulas of higher order to compute the vorticity field on the boundary that are both stable and consistent and permit highly accurate computations. In the integral formulation approach the equations to be solved are complemented with a boundary condition that may involve the boundary value of the vorticity (Wu and Thompson (1973), Chorin (1973)) or the vorticity flux (Kinney and Cielak (1974)).

In the present scheme we follow the integral formulation and employ the model of vorticity creation originally devised by Lighthill (1963) and further formulated by Kinney and his coworkers (Kinney and Paolino (1974), Kinney and Cielak (1974), Schmall and Kinney (1974), Taslim, Kinney and Paolino (1984), Hung and Kinney (1988)). This formulation is considered in the context of vortex methods and accounts for vorticity generation at the solid walls due to the no-slip condition. Unlike previous vortex schemes, *no new particles are generated* in the vicinity of the boundary, but the strength of the existing ones are modified (based on the vorticity flux at the wall) so that the 'no-slip' condition is enforced.

5.2 Vorticity Creation at a Solid Wall - Lighthill's Model

The basis of the present formulation was originally proposed by Lighthill (1963). The key observation is that once the vorticity field is known then the entire flow field may be determined (via the Biot-Savart law of velocity induction). The vorticity field is convected and diffused in the fluid but in the presence of solid boundaries one has to account for the vorticity production on the solid walls as well. Lighthill models this vorticity creation process by considering the body surface as a collection of vorticity sources and sinks. To calculate these (unknown) vorticity strengths he proposes that the velocity field must be computed on the solid boundary from the known vorticity

field. From kinematic considerations in order to ensure the no-through flow boundary condition a potential flow correction needs to be superimposed to this velocity field. The resulting velocity field would have (in general) a non-zero tangential velocity component and one may view the surface of the body as a vortex sheet. To ensure the no-slip condition and model the vorticity creation process on the solid boundary this vortex sheet has to be related to the vorticity production at the wall. Lighthill concludes the description of his model by stating that the vorticity per unit area has been created and is equal to the negative of this vortex sheet strength. What remains incomplete in this model is how this vorticity enters the fluid adjacent to the wall or how the vortex sheet strength may be incorporated in a vorticity type boundary condition.

One may observe that the strength of the vortex sheet has dimensions of velocity (or length over time). To obtain an appropriate (dimensionally correct) vorticity boundary condition this vortex sheet strength can be manipulated so that a Dirichlet type (vorticity with dimensions 1/Time) or Neumann type (vorticity flux with dimensions of acceleration) may be obtained. This is basically the point of diversion of the various formulations involving vorticity boundary conditions. Chorin (1973) divides the strength of the vortex sheet by a length equal to the elementary discretization length on the body surface, whereas Wu (1976) divides it by the distance from the wall to the first mesh point in the computational domain, to obtain the vorticity on the body. Kinney and his coworkers envision this vortex sheet as equivalent to a vorticity flux over a small time interval (thus dividing the sheet strength by time to obtain units of acceleration). An integral constraint is imposed on all formulations on the vorticity created at the wall so as to satisfy Kelvin's theorem of production of circulation.

As it is evident from the above discussion there is not a unique and mathematically rigorous formulation relating the strength of the vortex sheet to the vorticity boundary conditions. The jury is still out as to whether Dirichlet or Neumann type conditions are more appropriate. It is interesting to include here some statements to this regard from Hung and Kinney (1988) :

> "It seems clear that a great deal of commonality, as well as diversity, exists between those various treatments of boundary conditions. It is perhaps disappointing that even in the light of such evidence, one cannot still resolve rigorously such a fundamental issue as to whether Dirichlet or Neumann boundary conditions are the correct ones. There are nevertheless strong indications that whatever

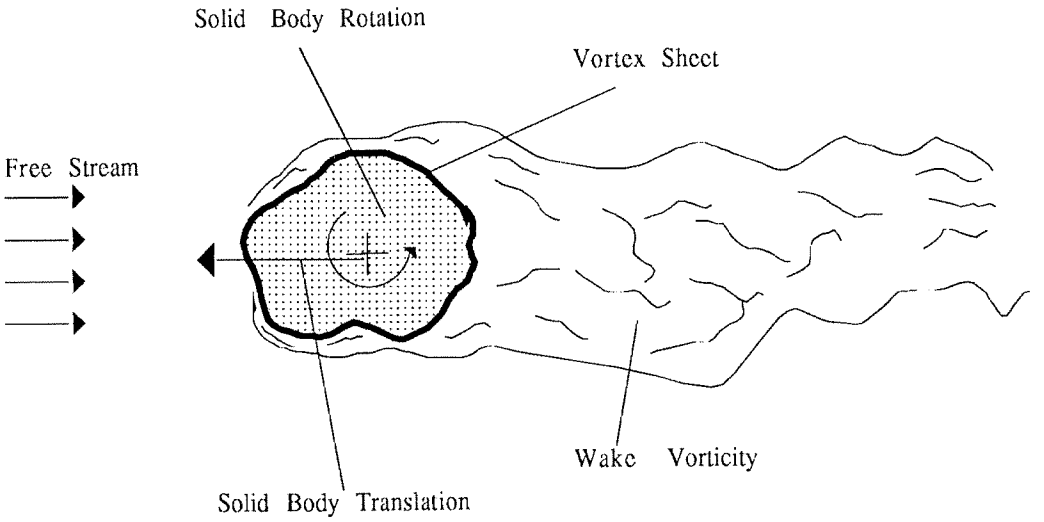


FIG. 5.1 Sketch showing the different contributions to the flow field of a body in translation and rotation immersed in a viscous incompressible flow field.

the precise form may be, they must be integral in character rather than local. Furthermore, it is the opinion of the present authors that theses must involve Neumann boundary conditions, since these occur most naturally in practice. It is unlikely that numerical experiments will ever resolve this question, and perhaps it is not important to do so. Clearly, plausible and even excellent results can be obtained by any of the several methods discussed when used with consistent and convergent numerical formulations on good grids. For that reason, perhaps these should be classified as models for the vorticity production which occurs at solid boundaries rather than rigorous mathematical constraints or boundary conditions."

In the present work the Neumann type vorticity boundary condition was chosen. This choice was mainly dictated by the use of vortex methods for the resolution of the vorticity transport equation. A Dirichlet type condition would explicitly involve in the scheme the wall vorticity. The computation of such a quantity is prone to interpolation errors that are further augmented by the use of a Lagrangian grid. Moreover this vorticity field needs to be discretized with blobs of finite core size. The existence of such blobs on the boundary introduces a physically absent smoothing region for the vorticity region on the boundary that significantly increases the numerical diffusion of the scheme. Moreover according to Chorin's model this would imply a continuous increase to the number of computational elements that enter the fluid at each time

step thus increasing the cost of the computations. On the other hand the present use of a Neumann type condition does not require additional computational elements (vortex blobs) on the surface of the body. Based on the implementation presented here in this vorticity flux is distributed by diffusion to the existing blobs thus altering their strength but without increasing their population. The technique presented here in is also consistent with the scheme of Particle Strength Exchange (PSE) for the diffusion of vorticity in the flow field as all viscous effects in the computations are resolved via the modification of the strengths of the vortex particles.

5.2.1 Mathematical Formulation

We consider a body translating with velocity $\mathbf{U_b}(t)$ and rotating with angular velocity ($\Omega(t)$) immersed in a flow field induced by a uniform flow ($\mathbf{U}_\infty$) and vorticity in the wake ($\omega_f(\mathbf{x},t)$). In the present formulation the solid body is represented by suitable vorticity distributions determined so that the entire flow field follows the prescribed solid body motion. The interior of the body is replaced by a uniform vorticity field (ω_b) with strength equal to twice the magnitude of the angular velocity ($\omega_b = 2\Omega$), while the surface of the body is replaced by a vortex sheet (bound vorticity) with strength $\gamma(s)$ (Fig. 5.1).

As discussed by Lamb, ((1932), Ch.iii) the kinematic velocity field is uniquely determined from the vorticity field if the no-through flow boundary condition ($(\mathbf{u} - \mathbf{U_s}) \cdot \mathbf{n} = 0$) has been enforced on the surface of the body. A vortex sheet appears then on the surface of the body and the enforcement of the no-through flow boundary condition is equivalent to determining the strength of this vortex sheet. The strength of the vortex sheet is computed using the streamfunction of the flowfield as discussed in Section 5.3. The resulting integral equation is given by :

$$\gamma(s) - \frac{1}{\pi} \oint \frac{\partial}{\partial n} \left[Log|\mathbf{x}(s) - \mathbf{x}(s')| \right] \gamma(s')\, ds' = -2\, h(\mathbf{x}(s)) \tag{5.6}$$

where :

$$h(\mathbf{x}(s)) = \frac{\partial \Psi_f}{\partial n}(\mathbf{x}(s)) + \frac{\partial \Psi_\Omega}{\partial n}(\mathbf{x}(s)) - \mathbf{U_s} \cdot \mathbf{n} \tag{5.7}$$

The solution properties of the above equation were originally studied by Prager (1928) who first introduced the concept of replacing the body surface by a continuous vortex sheet. Equation 5.10 is singular as it admits a non-unique solution and requires that an extra constraint be imposed on the strength of the vortex sheet. This property of the equation is a fortuitous result however, as it allows for the coupling of the kinematic description of the flow field with the viscous wall production of vorticity.

The model presented herein relies in the nullification of this spurious vortex sheet at the body surface so as to enforce the no-slip boundary condition. As this vortex sheet is a constituent of the flow field its strength should account for the modification of the circulation of the flow field. Hence when it is eliminated from the body surface in the interval $[t,\, t+\delta t]$ the circulation (Γ) of the flow field would be modified according to:

$$\oint \gamma(s)\, ds \;=\; \int_t^{t+\delta t} \frac{\mathrm{d}\Gamma}{\mathrm{d}t'}\, dt' \tag{5.8}$$

On the other hand by Kelvin's theorem the rate of change of circulation in the flow field is defined as:

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}t} \;= \nu \oint \frac{\partial \omega}{\partial n}(s)\, ds \;=\; -2\,\frac{\mathrm{d}\Omega}{\mathrm{d}t}\, A_B \tag{5.9}$$

where A_B is the area of the body. Integrating now Eq. 5.9 in the interval $[t,\, t+\delta t]$ we obtain that :

$$\int_t^{t+\delta t} \frac{\mathrm{d}\Gamma}{\mathrm{d}t'}\, dt' \;=\; \int_t^{t+\delta t} dt' \oint \nu \frac{\partial \omega}{\partial n}(s)\, ds \;=\; -2\, A_B\, [\Omega(t+\delta t) \,-\, \Omega(t)] \tag{5.10}$$

Comparing Eq. 5.8 and Eq. 5.10 the strength of the vortex sheet may be related to the vorticity flux at the body surface as:

$$\boxed{\nu \int_t^{t+\delta t} \frac{\partial \omega}{\partial n}(s)\, dt' \;=\; -\gamma(s)} \tag{5.11}$$

or if we consider this vorticity flux to be constant over the small interval of time (δt)

$$\nu\, \frac{\partial \omega}{\partial n}(s) \;=\; -\gamma(s)/\delta t \tag{5.12}$$

This constitutes then a Neumann type vorticity boundary condition equivalent to the no-slip boundary condition as expressed by Eq. 5.1. The above formulation allows then to impose an integral constraint on the strength of the vortex sheet γ

$$\boxed{\oint \gamma(s)\,ds \;=\; -2\,A_B[\Omega(t+\delta t) \;-\; \Omega(t)]} \tag{5.13}$$

and provides a closure for Eq. 5.6 that admits now a unique solution. *

A few observations should be made here as to the behaviour of the above formulation in the limit of $\delta t \rightarrow 0$. First note that at the end of a time step the spurious vortex sheet has been eliminated. The strength of the vortex sheet is dependent on the external flow field and at the limit of a vanishing δt, there would be accordingly a vanishingly small change for γ as well so that the vorticity flux would remain finite. The numerical vorticity flux should be consistent with the actual vorticity flux as computed by applying the momentum equation on the wall. In body fitted coordinates this results in

$$\mathbf{s} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}|_{\text{wall}} \;=\; -\frac{1}{\rho}\frac{\partial p}{\partial s}|_{\text{wall}} \;+\; \nu\frac{\partial \omega}{\partial n}|_{\text{wall}} \tag{5.14}$$

Vorticity is generated in the fluid due to the tangential component of the pressure gradient and a possible acceleration of the body surface. In the present fractional step algorithm (Ch.2) this pressure gradient is manifested by a spurious slip velocity observed on the body surface. We may consider this slip velocity as an acceleration 'equivalent' to a vorticity flux generated at the wall, so that at each time step (δt) Eq. 5.11 is satisfied.

Once γ has been computed (solving Eq. 5.13 and Eq. 5.6) the vorticity flux is determined at the surface of the body according to Eq. 5.11. This vorticity flux is subsequently distributed to the particles (by appropriately modifying their strength) as described in Section 5.4 so that the spurious slip velocity is nullified and vorticity is generated in the fluid. This technique of enforcing the no-slip boundary condition is consistent with the scheme of PSE (Ch.4) that accounts for diffusion. In the present method all viscous effects are resolved by appropriately modifying the strength of the particles.

---

* Note that in Hydrodynamics the equivalent constraint would be a Kutta type condition

5.3 Computation of the Surface Vortex Sheet (γ)

This section deals then with the computation of the potential part of an unsteady, incompressible, viscous flow around an arbitrary configuration.

The boundary may be considered as a surface of discontinuity (vortex sheet) (Prager, 1928) and boundary elements (panels) may be used for its discretization. The inviscid boundary condition is then applied in a Neumann or Dirichlet form on the streamfunction, to determine the unknown vorticity distribution. However the solution to this problem is not unique and the additional constraint of conservation of total circulation needs to be imposed. The resulting sets of equations are not numerically well conditioned and the accuracy of the solution deteriorates as the thickness of the body is decreased and/or the number of the panels increases. Here a rigorous approach is presented (similar to the one used in Baker and Shelley, 1986) involving the application of the internal Neumann boundary condition, in which the resulting system of equations is well conditioned, permitting an efficient and accurate solution with direct or iterative matrix inversion techniques. The method does not increase the computational cost and it is found to improve the conditioning of the system by several orders of magnitude, the improvement being more pronounced as the number of panels increases.

The streamfunction in any location of the domain ($\mathbf{x}$) is expressed at any instant, as the linear superposition of the streamfunctions induced by the constituent vorticity fields of the present formulation as :

$$\Psi(\mathbf{x}) \;=\; \Psi_{\gamma}(\mathbf{x}) \;+\; \Psi_{\Omega}(\mathbf{x}) \;+\; \Psi_{\mathrm{f}}(\mathbf{x}) \tag{5.15}$$

where :

$$\Psi_{\gamma}(\mathbf{x}) \;=\; -\frac{1}{2\pi} \oint Log|\mathbf{x}-\mathbf{x}(s')|\,\gamma(s')\,ds' \tag{5.16}$$

is the streamfunction induced by the bound vorticity (vortex sheet),

$$\Psi_{\Omega}(\mathbf{x}) \;=\; -\frac{\Omega(t)}{\pi} \int_{B} Log|\mathbf{x}-\mathbf{x}'|\,d\mathbf{x}' \tag{5.17}$$

is the streamfunction induced by the solid body rotation (i.e. the vorticity field interior to the body, B), and finally

$$\Psi_{\mathrm{f}}(\mathbf{x}) \;=\; -\frac{1}{2\pi} \int_{A} \omega_f(\mathbf{x}')Log|\mathbf{x}-\mathbf{x}'|\,d\mathbf{x}' \;+\; (\mathbf{U}_{\infty}(\mathbf{t}) \times \mathbf{x}) \cdot \widehat{\mathbf{e}_z} \tag{5.18}$$

is the streamfunction induced by the vorticity in the wake (region A) and the free stream velocity.

In order to determine the flow uniquely the strength of the surface vortex sheet needs to be evaluated. This may be achieved by enforcing a Dirichlet or a Neumann type boundary condition for the streamfunction of the flow. Taking into account the additional constraint that the total circulation ejected from the surface of the body to the fluid is equal to the respective change in circulation inside the body ($\delta\Gamma_B$) results in :

$$c = -\frac{1}{2\pi}\oint Log|(\mathbf{x}(s)-\mathbf{x}(s')|\,\gamma(s')\,ds' + \Psi_f(\mathbf{x}(s)) + \Psi_\Omega(\mathbf{x}(s)) + \left((\mathbf{U}_\infty - \mathbf{U_b})\times\mathbf{x}(s)\right)\cdot\widehat{\mathbf{e}_z} - \Omega\,|\mathbf{x}(s)-\mathbf{x}_b|^2 \tag{5.19a}$$

$$\oint \gamma(s')\,ds' + \delta\Gamma_B = 0 \tag{5.19b}$$

for the Dirichlet type boundary condition and in :

$$\gamma(s) - \frac{1}{\pi}\oint \frac{\partial}{\partial n}\left[Log|\mathbf{x}(s)-\mathbf{x}(s')|\right]\gamma(s')\,ds' = -2\,h(\mathbf{x}(s)) \tag{5.20a}$$

$$\oint \gamma(s')\,ds' + \delta\Gamma_B = 0 \tag{5.20b}$$

for the Neumann type boundary condition (with $h(\mathbf{x}(s))$ given by Eq. 5.7). As discussed in the previous section the change of circulation of the wake at any instant is defined as :

$$\delta\Gamma_B = 2\,A_B\,[\Omega(t+\delta t) - \Omega(t)] \tag{5.21}$$

where A_B denotes the area of the body.* These sets of equations may be solved to any accuracy by a panel method. Discretizing the body with M vortex panels results in a system of equations for the M unknown strengths. The two linear sets of equations may be expressed in matrix form :

$$Kf = g \quad \text{(Dirichlet B.C.)} \quad ; \quad Gf = h \quad \text{(Neumann B.C.)}.$$

---

* In numerical computations the circulation of the wake, as computed by the linear superposition of the circulations of the vortex elements, would not equal the theoretical value given by Eq. 5.21. For example in actual computations around a non-rotating body the computed circulation would not be zero but would have some small value that needs to be accounted for in $\delta\Gamma_B$ in order to obtain conservative computations.

System $Kf = g$ has M+1 equations with M+1 unknowns (the M+1th unknown being the constant c). However for thin bodies or bodies with cusped trailing edges two panels can be very close to each other while corresponding to opposite normals. This would result in the matrix K having two nearly identical rows and therefore being nearly singular. System $Gf = h$ has M+1 equations but with only M unknowns, the strengths of the panels. In order to solve this latter system of equations several approaches are plausible (Hess, (1975, 1990)): source method, least squares solution, introduction a new unknown, elimination of one equation etc.. However these approaches rely mainly on empirical criteria which become more important than any details of the numerical implementation. Moreover in the case of thin bodies , two panels can be separated by a distance much smaller than either length (e.g., in high curvature trailing edges) resulting in the existence of two nearly equal but opposite signed rows in the matrix G producing an ill-conditioned system of equations (Morchoinse et. al., (1987)). In practice this anomaly is more pronounced for source methods because the solution source strength increases without bounds as the body thickness goes to zero. Our method consists of an application of the Fredholm alternative to the solution of Eq. 5.20. This results in a system of equations which is well behaved, independent of the thickness of the body and the size of the panels and therefore improving the accuracy of the calculations especially when large numbers of panels are necessary for the description of the body.

5.3.1 Mathematical Formulation

By setting

$$G(s,s') = \frac{1}{\pi}\frac{\partial}{\partial n}\left[Log|\mathbf{x}(s) - \mathbf{x}(s')|\right]$$

then Eq. 5.20 may be expressed equivalently as :

$$\gamma(s) - \oint G(s,s')\,\gamma(s')\,ds' = h(s) \tag{5.22}$$

The Eq. 5.22 is an integral equation of the second kind with a nontrivial homogeneous solution (Prager (1928), Martensen (1959)). According to the Fredholm alternative (Mihklin, 1964) a solution to Eq. 5.22 exists if

$$\oint h(s')\,\psi(s')\,ds' = 0$$

where $\psi(s)$ is the solution of the eigenvalue problem for the transposed equation with $\lambda = 1$, i.e.,

$$\oint G(s', s)\, \psi(s')\, ds' = \lambda \psi(s) \tag{5.23}$$

For the kernel $G(s, s')$ considered $\psi(s) = const.$ is the solution for $\lambda = 1$ so the necessary condition is:

$$\oint h(s')\, ds' = 0$$

The above condition is equivalent to:

$$\oint \mathbf{u}_o \cdot d\mathbf{l} = \oint \left[\mathbf{U_b} + \Omega(t)\, \widehat{\mathbf{e}_z} \times (\mathbf{x}_s - \mathbf{x}_b)\right] \cdot d\mathbf{l}$$

(with $\mathbf{u}_o = \nabla \times \left((\Psi_f + \Psi_\Omega)\hat{\mathbf{e}}_z\right)$), which is indeed the case. The solution of Eq. 5.22 is not unique since if $\gamma_0(s)$ is a solution then an arbitrary number of solutions $\gamma(s)$ of the form:

$$\gamma(s) = \gamma_0(s) + \alpha\, \phi(s)$$

may be obtained, where α is an arbitrary constant and $\phi(s)$ the solution of the following eigenvalue problem for $\lambda = 1$,

$$\oint G(s, s')\, \phi(s')\, ds' = \lambda \phi(s) \tag{5.24}$$

However uniqueness of the solution for our problem is guaranteed by the the application of Kelvin's theorem.

$$\oint \gamma(s)\, ds = -\delta\Gamma_B \tag{5.25}$$

or equivalently α is uniquely determined by :

$$\alpha = -\delta\Gamma_B + \frac{\oint \gamma_0(s)\, ds}{\oint \phi(s)\, ds} \tag{5.26}$$

Note here that for **multiple bodies** Kelvin's theorem should be enforced for each individual body and its corresponding circulation.

5.3.2 Proposed Scheme

An alternative way for the application of the inviscid boundary condition is proposed here based on the spectral decomposition of the kernel $G(s,s')$. This kernel may be decomposed as :

$$G(s,s') = \sum_{i=1}^{\infty} \lambda_i \phi_i(s) \psi_i(s') \tag{5.27}$$

with $\lambda_i, \phi_i(s)$, eigenvalues and eigenfunctions given by the solution of Eq. 5.24 and $\psi_i(s)$ eigenfunctions of the transposed kernel given by the solution of Eq. 5.23. These eigenfunctions are normalized so that equations Eq. 5.23 and Eq. 5.24 are valid for the decomposed kernel defined in Eq. 5.27. Hence :

$$\oint \phi_i(s)\, \psi_j(s)\, ds = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases} \tag{5.28a}$$

and

$$\psi_1(s) \;=\; \text{const.} \;=\; 1/L \tag{5.28b}$$

where L is the perimeter of the shape considered. Note that the eigenfunction $\phi_1(s)$ corresponding to the eigenvalue $\lambda_1 = 1$ is a solution of Laplace's equation satisfying the inviscid boundary condition in the absence of any external flow field.

Now the condition imposed by Kelvin's theorem may be substituted by an equivalent one, by multiplying both sides of Eq. 5.25 with $\lambda_1\, \phi_1(s)\, \psi_1(s')$, so that the new set of equations is :

$$\gamma(s) - \oint G(s,s')\gamma(s')\, ds' = h(s) \tag{5.29a}$$

$$\frac{\phi_1(s)}{L} \oint \gamma(s')\, ds' \;=\; -\frac{\phi_1(s)}{L}\, \delta\Gamma_B \tag{5.29b}$$

Adding Eq. 5.29b to Eq. 5.29a results in the final form of the integral equation that needs to be solved for the unknown function $\gamma(s) = d\Gamma/ds$:

$$\gamma(s) - \oint \left[G(s,s') - \frac{\phi_1(s)}{L}\right]\gamma(s')\, ds' = h(s) - \frac{\phi_1(s)}{L}\, \delta\Gamma_B \tag{5.30}$$

In this form the new kernel is just the kernel given by Eq. 5.27 but with the first term in the spectral decomposition eliminated. Hence the singularity associated with the kernel $G(s, s')$ is annihilated and Eq. 5.30 is well-posed. This equation is solved using panel methods and the resulting system of equations is a well conditioned one. Note that in the above formulation of the problem it is necessary to know a priori the form of the eigenfunction $\phi_1(s)$, which is not available in general for arbitrary shapes. For elliptic bodies (including the cylinder and the flat plate) this eigenfunction was determined analytically and is given as a function of the eccentricity ϵ of the ellipse and the polar angle θ :

$$\phi_1(\theta) \;=\; (1 - \epsilon\cos^2(\theta))^{-1/2}$$

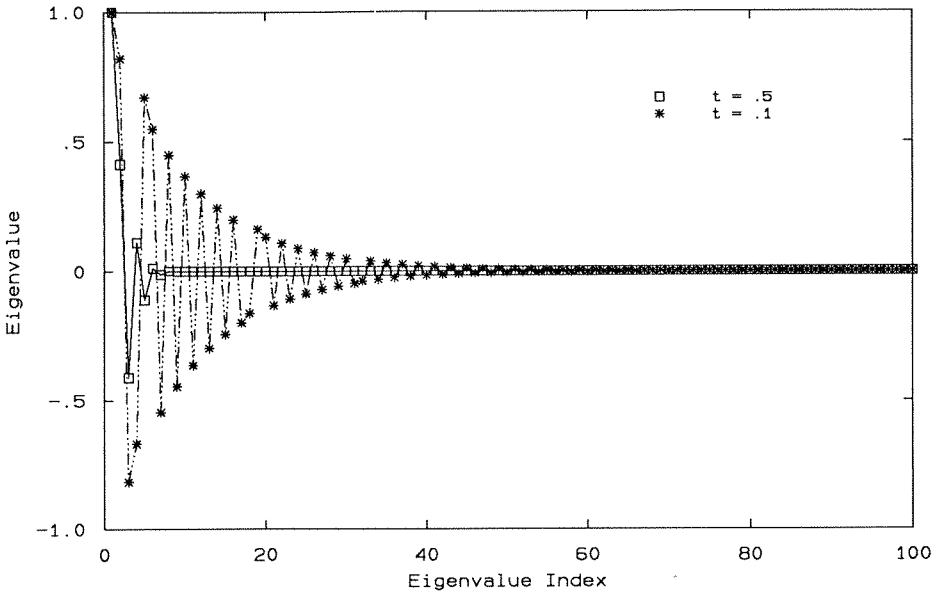


FIG. 5.2 Eigenvalues of ellipses having thickness (t): 50%, 10%.

For arbitrary configurations (e.g., a NACA0012 airfoil), $\phi_1(s)$ may be obtained by solving the eigenproblem (Eq. 5.24) using panel methods. The eigenfunction is then obtained using as collocation points the eigenvector of the resulting matrix. This adds to the computational cost of the method, but this eigenfunction needs to be computed only once for the considered shape. For the case of a cylinder (of radius R), $\epsilon = 0$, so the unknown function f(s) may be obtained directly as (Eq. 5.30) reduces to :

$$\gamma(s) = h(s) - \frac{1}{2\pi \mathrm{R}}\,\delta\Gamma_B \tag{5.31}$$

In Fig. 5.2 the eigenvalues for two ellipses are shown. Calculations for various ellipses exhibit an interesting oscillatory behavior of the eigenvalues and a broadening of the spectrum as the thickness of the body decreases.

5.3.3 Numerical Application - Results

In order to compare the various methods for the application of the boundary conditions the possibility of ill-conditioning in the numerical solution to the resulting linear systems of equations was examined. This ill-conditioning is in the sense that the results of the computation depend continuously on the data with a proportionality constant that is not very large (Isaacson and Keller, 1966) That is, considering the system $Ax = b$, small changes in b result in large changes in x. Conversely the underlying requirement for a well-conditioned system is that a small perturbation ϵ of the position of the vortices in the wake of the body resulting in a small change δb on the right hand side, should not result in a significant change δx on the results of the computation. As a measure of this change we consider the condition index κ of the matrix A defined as :

$$\kappa = \kappa(A) = ||A^{-1}||\ ||A||$$

so that

$$\frac{||\delta x||}{||x||} \leq \kappa \frac{||\delta b||}{||b||}$$

In the present calculations of κ the 1-norm has been implemented.

In Fig. 5.3 and Fig. 5.4, the condition index for the system resulting from the application of the Dirichlet boundary condition and the condition index of the system resulting from the proposed scheme are shown for the potential flow around ellipses of different thickness and a NACA0012 airfoil, respectively

As can be seen from these figures, as the number of panels is increased and the thickness of the body is decreased the Dirichlet boundary condition results in a system that is ill-conditioned. For a small number of panels , using computers with 8 or 10 significant figures this anomaly is not seriously affecting the results. When large numbers (in the order of thousands) of panels are necessary for the description of the boundary this ill-conditioning becomes important.

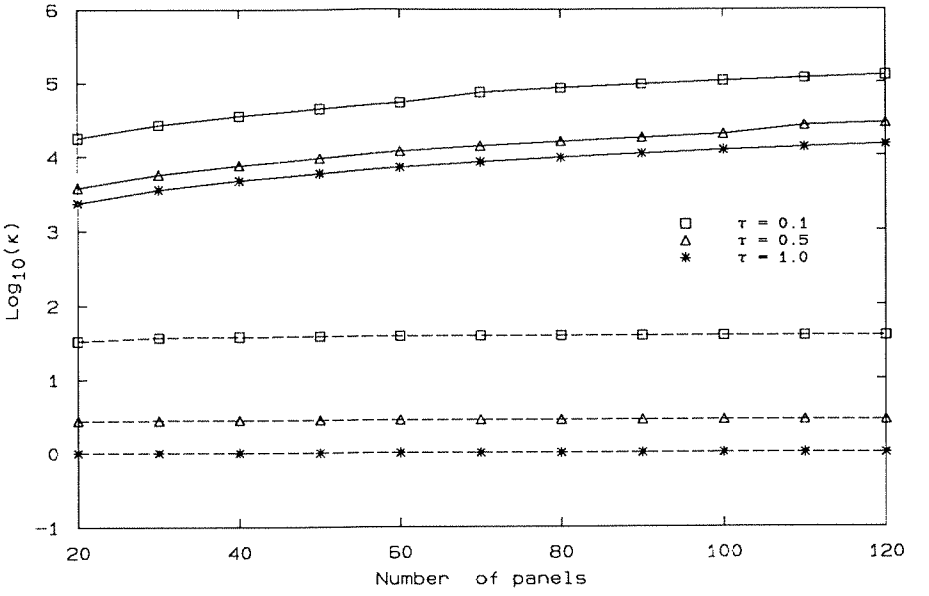


FIG. 5.3 Condition Index resulting from : the classical approach (solid lines) and the proposed method (dashed lines) for ellipses of different thickness(τ).

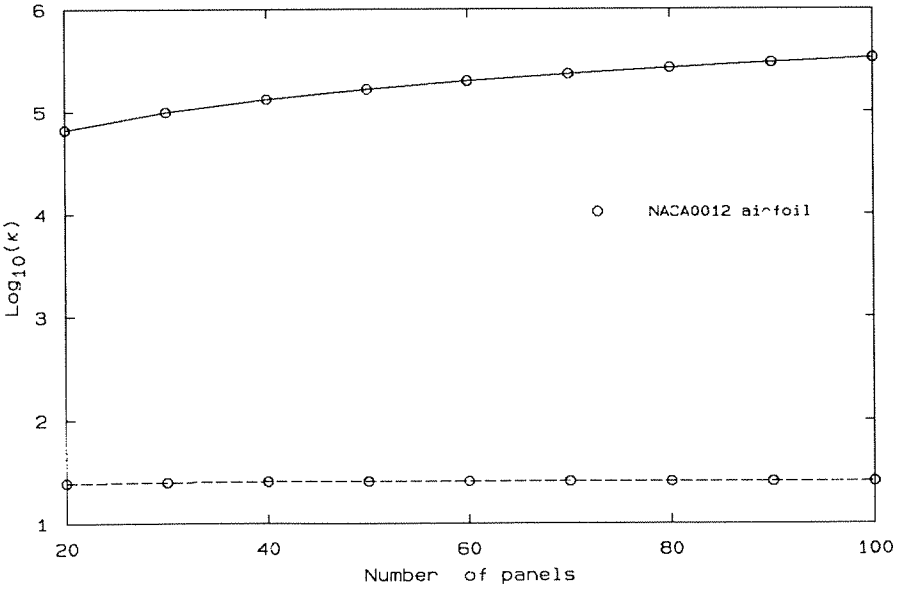


FIG. 5.4 Condition Index resulting from : the classical approach (solid line) and the proposed method (dashed line) for a NACA0012 airfoil.

For example, in the context of using vortex methods for the description of the flow,

in order to obtain accurate results one must always keep the ratio of the distance of the vortex from the panel to the length of the panel less than unity. So when boundary layers are resolved with vortices, their small distance from the body dictates the use of very large numbers of panels. However this would result in higher computer cost and would deter the accuracy of the solution. On the other hand the application of the present scheme results in a very well conditioned system of equations that tends asymptotically to a small value as the number of panels is increased.

The advantage of the present formulation is more pronounced as the number of the panels is increased and/or the thickness of the body decreases. When we solve the system of equations by direct elimination the conditioning (and the size) of the system is such that it does not pose any numerical difficulty. However as the number of unknowns increase direct methods become prohibitive and we have to rely on iterative techniques such as conjugate gradient. The conditioning of the coefficient matrix is a key factor to the convergence of these techniques and as it may be observed from the above figures, the present formulation is advantageous for such computations.

5.3.4 Computation of the Tangential Velocity on the Body

In the present scheme a key element for the enforcement of the boundary condition is the computation of the tangential velocity on the surface of the body. In computations with vortices this quantity is usually prone to numerical errors and noisy results appear when the distance of the vortices to the panel is smaller than the panel size.

To alleviate this difficulty the tangential component of the velocity on the boundary due to the free stream and the vortices in the wake is computed using the potential ($\mathbf{u} = \nabla\Phi$) of the flow field. The use of the potential to compute the slip velocity on the body ensures that the flow field remains divergence free throughout the domain. The average slip velocity between two locations on the body is:

$$(\bar{U}_s)_{ab} = \frac{1}{\delta s}\int_{s_a}^{s_b} \frac{\partial \Phi}{\partial s}\, ds$$

So a panel having size d_p between the points i and $i+1$ of the boundary would experience an average tangential velocity equal to :

$$(\bar{U}_s)_i = \frac{\Phi_{i+1} - \Phi_i}{d_p}$$

Note here that the potential induced by a vortex (of strength Γ_v and located at (x_v, y_v)) at a point (x,y) is computed* as :

$$\Phi(x,y) = \frac{\Gamma_v}{2\pi}\mathrm{TAN}^{-1}(\frac{y_v - y}{x_v - x})$$

The use of this average velocity allows for accurate computations without having to worry about the distance of the vortices from the body surface. The potential field due to the vortices is computed by a linear superposition of their potentials. This implies that if we discretize the body with M panels the cost of computing the influence of N particles in the wake would be $\mathcal{O}(MN)$. This cost can be prohibitive for large numbers of particles as the slip needs to be computed at each time step. In the present scheme a fast algorithm (similar to the one presented in Ch.2) is implemented that uses the multipole expansions of the vortices in the domain. Direct interactions are computed for the small fraction of n_{near} vortices that are close to the boundary and multipole expansions are computed for those that are sufficiently removed from it. This results in a scheme that scales as $\mathcal{O}(M\, n_{\mathrm{near}} + M \log N)$ so that efficient computations are possible.

5.4 Distribution of the Vorticity Flux

Once the vorticity flux has been computed (via Eq. 5.14) it has to be distributed to the particles in the domain so that vorticity enters the fluid. As it was described in Chapter 1 this is achieved in the context of a fractional step algorithm by the solution of a diffusion equation with homogeneous initial conditions and a Neumann boundary condition.

---

* A note should be noted here as to the computational difficulty when employing the ATAN2 function in the computer. When the branch cut of a vortex intercepts a panel then the coordinates of the vortex should be rotated by 180^o so as to account for the average tangential velocity properly.

5.4.1 Mathematical Formulation

We consider the diffusion equation for the vorticity $\omega(\mathbf{x},t)$ with homogeneous initial conditions and boundary conditions of the Neumann type:

$$\begin{aligned}
\omega_t - \nu\nabla^2\omega &= 0, \qquad \text{in} \quad \mathcal{D}\times[t-\delta t,t] \\
\omega(\mathbf{x},t-\delta t) &= 0 \qquad \text{in} \quad \mathcal{D} \\
\frac{\partial\omega}{\partial n} &= F(\mathbf{x},t) \qquad \text{on} \quad \partial\mathcal{D}\times[t-\delta t,t].
\end{aligned}$$

where $\mathcal{D}$ denotes the computational domain bounded by the surface of the body ($\partial\mathcal{D}$). The solution of the above equation may be expressed in integral form (Friedman, 1964) as :

$$\omega(\mathbf{x},t) = \int_{t-\delta t}^{t}\int_{\partial\mathcal{D}} G(\mathbf{x},t;\boldsymbol{\xi},\tau)\,\mu(\boldsymbol{\xi},\tau)\,ds_\xi\,d\tau \tag{5.32}$$

where $\mu(\mathbf{x},t)$ is determined by the solution of :

$$-\frac{1}{2}\mu(\mathbf{x},t) + \int_{t-\delta t}^{t}\int_{\partial\mathcal{D}} \frac{\partial G}{\partial n}(\mathbf{x},t;\boldsymbol{\xi},\tau)\,\mu(\boldsymbol{\xi},\tau)\,ds_\xi\,d\tau = F(\mathbf{x},t) \tag{5.33}$$

with :

$$G(\mathbf{x},t;\boldsymbol{\xi},\tau) = \frac{1}{4\,\pi\,\nu\,(t-\tau)}\exp\left(\frac{|\mathbf{x}-\boldsymbol{\xi}|^2}{4\,\nu\,(t-\tau)}\right)$$

The resulting expressions for the vorticity field involve integrals only over the surface of the body. Those integrals may be discretized with a boundary integral method by assuming the surface of the body as being composed by a set of discrete panels (straight or curved) and assuming a certain variation (constant, linear, etc.) of the unknown function $\mu(\mathbf{x},t)$ in space (over the panels) and time. For the present formulation we assume that $\mu(\mathbf{x},t)$ remains constant in time (for the small interval δt in order to facilitate the evaluation of these integrals.

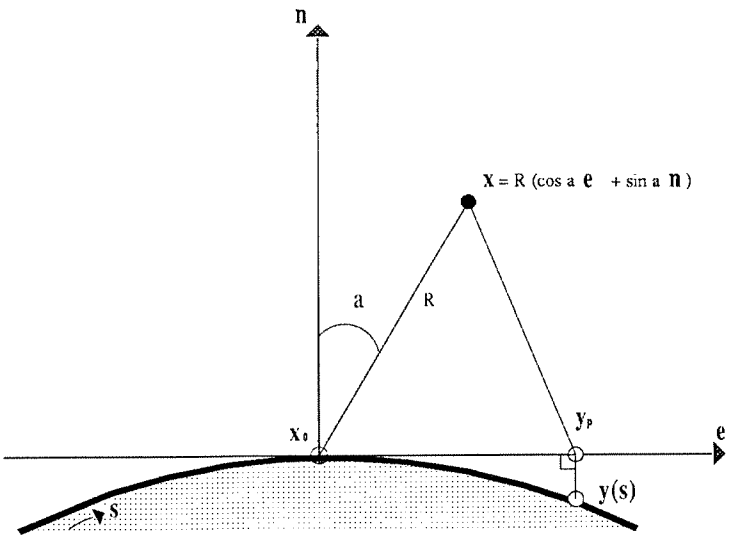


FIG. 5.5 Definition Sketch.

5.4.2 Geometrical Definitions

In order to explicitly evaluate the above integral let us consider the geometric representation of the body. From Fig. 5.5 points on the body are defined by:

$$\mathbf{y} = \mathbf{x_0} + x_p\widehat{\mathbf{e}} - \frac{x_p^2}{2\rho}\widehat{\mathbf{n}} + \mathcal{O}(x_p^3),$$

where ($\rho = 1/\kappa$) the local radius of curvature of the body. Points in the domain are defined as:

$$\mathbf{x} = \mathbf{x_0} + R\sin(a)\widehat{\mathbf{e}} + R\cos(a)\widehat{\mathbf{n}}$$

Based on the above definitions we have also that:

$$d\mathbf{y} = dx_p\widehat{\mathbf{e}} - \frac{x_p}{\rho}dx_p\widehat{\mathbf{n}}$$

and the distance **body-point** is then:

$$\| \mathbf{x} - \mathbf{y} \|^2 = R^2 - 2R\sin(a)\,x_p + (1 + \frac{R\cos(a)}{\rho})x_p^2 + \mathcal{O}(x_p^3)$$

The evaluation of the integrals in Eq. 5.32 and Eq. 5.33 is further simplified if we choose to neglect the terms associated with the local curvature of the body or equivalently describe the body with tangential flat panels. A panel approximation to the body (Fig. 5.5) then introduces points

$$\mathbf{y_p} = x_p\widehat{\mathbf{e}} + \mathbf{x_0}$$

with:

$$d\mathbf{y_p} \;=\; dx_p \widehat{\mathbf{e}}$$

and the distance from the field points to the body may be approximated by the distance **panel-point**

$$\parallel \mathbf{x} \;-\; \mathbf{y_p} \parallel^2 \;=\; R^2 \;-\; 2R\sin(a)\, x_p \;+\; x_p^2$$

5.4.3 Vorticity Evaluation

Substituting now the above approximations for the surface of the body in Eq. 5.32 we have the following representation for the vorticity field at point **x**:

$$\omega(R,a) \;=\; \int_{t-\delta t}^{t} \frac{1}{4\pi\nu(t-\tau)} \oint_{\partial\mathcal{D}} e^{\frac{-R^2 \,+\, 2R\,\sin(a)\,x_p \,-\, x_p^2}{4\nu(t-\tau)}}\, \mu_0(x_p)\, dx_p\, d\tau$$

Assuming furthermore a constant strength μ_i for the heat potential over each panel (of size $2d$) we may alternatively express the above equation as :

$$\omega \;=\; \sum_{i=1}^{M} \mu_i I_i \;=\; \sum_{i=1}^{M} \omega_i$$

where

$$I_i \;=\; \int_{t-\delta t}^{t} \frac{e^{\frac{-R^2}{4\nu(t-\tau)}}}{4\pi\nu(t-\tau)}\, d\tau \int_{-d}^{d} e^{\frac{2R\,\sin(a)\,x_p \,-\, x_p^2}{4\nu(t-\tau)}}\; dx_p$$

The integral over the panel may be calculated explicitly so that the vorticity field induced by a panel, i, may be expressed as an integral over time :

$$\omega_i(\mathbf{x},t) \;=\; \frac{\mu_i}{2} \int_{t-\delta t}^{t} \phi(t-\tau)\, d\tau$$

with

$$\phi(t-\tau) \;=\; \frac{e^{\frac{-y^2}{4\nu(t-\tau)}}}{\sqrt{4\pi\nu(t-\tau)}} \left[\mathrm{erf}\left(\frac{d+x}{\sqrt{4\nu(t-\tau)}}\right) + \mathrm{erf}\left(\frac{d-x}{\sqrt{4\nu(t-\tau)}}\right)\right]$$

with $(x, y) = R(\sin(a), \cos(a))$ (Fig. 5.5). Taking the derivative of the last equation in respect with time we find:

$$\begin{aligned}\frac{\partial \omega_i}{\partial t} &= \frac{\mu_i}{2}[\int_{t-\delta t}^{t} \frac{\partial \phi(t-\tau)}{\partial t}\, d\tau + \phi(0) - \phi(\delta t)] \\ &= \frac{\mu_i}{2}[2\phi(0) - 2\phi(\delta t)]\end{aligned}$$

Now since $\phi(0) = 0$ for points away from the boundary , we obtain finally that

$$\begin{aligned}\frac{\partial \omega_i}{\partial t} &= -\mu_i\, \phi(\delta t) \\ &= \mu_i \frac{e^{\frac{-y^2}{4\nu\delta}}}{\sqrt{4\pi\nu\delta}} [\mathrm{erf}(\frac{d+x}{\sqrt{4\nu\delta}}) + \mathrm{erf}(\frac{d-x}{\sqrt{4\nu\delta}})]\end{aligned} \tag{5.34}$$

Using an Euler time stepping to integrate the above equation and assuming for the strength of the particles that $\Gamma_j = \omega(\mathbf{x}_j)\, h^2$

$$\Gamma_j^{n+1} = \Gamma_j^n - \sum_{i=1}^{M} \mu_i\, \delta t\, \phi(\delta t) h^2 \tag{5.35}$$

In the next section we present an efficient way to compute the values of the surface density μ_i on the panels by a fast evaluation of the double heat potential.

5.4.4 Evaluation of the Surface Density

To complete the evaluation of the vorticity field in the domain we need to compute the distribution of the surface density μ. Here Eq. 5.33 is solved explicitly by exploiting the local character of the Green's function G and its normal derivative on the body.

Consider the double layer heat potential which is defined as :

$$\mathcal{H}\mu(\mathbf{x}, t) = \oint_{\partial \mathcal{D}} \int_{t-\delta t}^{t} \frac{\partial G}{\partial n}(\mathbf{x}, t; \boldsymbol{\xi}, 0)\, \mu(\boldsymbol{\xi})\, d\boldsymbol{\xi}\, d\tau$$

If $\mathbf{s}$ is the coordinate along the boundary of the body, the body may be locally described as:

$$x = s \quad , \quad y = y(s)$$

Using a Taylor's expansion for the shape of the body we have that

$$\begin{aligned} y &= y(0) + y_s(0)s + \frac{1}{2}y_{ss}s^2 = \frac{\kappa}{2}s^2 \\ y_s &= y_s(0) + y_{ss}(0)s = \kappa s \\ \mu &= \mu(0) + \mu_s(0)s + \frac{1}{2}\mu_{ss}s^2 \end{aligned} \tag{5.36}$$

where $\kappa(s)$ is the local curvature of the body. The unit normal to the boundary is given by:

$$\mathbf{n} = \frac{[y_s(s), -x_s(s)]}{\sqrt{x_s^2 + y_s^2}} = \frac{[y_s(s), -1]}{\sqrt{1 + y_s^2}}$$

Differentiating the fundamental solution kernel we find that:

$$\nabla G = \frac{[-2(x-\xi), -2(y-\zeta)]}{4\nu(t-\tau)} G(\mathbf{x}, t; , \boldsymbol{\xi}, \tau)$$

Using the above expansions (Eq. 5.36) for the description of the curve and the variation of the surface density along it we obtain:

$$\mathcal{H}\mu = 2 \int_{t-\delta t}^{t} \oint_{\partial\mathcal{D}} \frac{e^{\frac{-s^2+y^2}{4\nu(t-\tau)}}}{4\pi\nu(t-\tau)} \frac{s\, y_s - y}{4\nu(t-\tau)} \mu(s)\, ds\, d\tau$$

To evaluate now the above integral we use the following transformations

$$z^2 = 4\nu(t-\tau), \quad s = z\, r.$$

so that $\mathcal{H}\mu$ becomes :

$$\mathcal{H}\mu = \frac{1}{\pi\nu} \int_{0}^{2\sqrt{\nu\delta}} \int_{-\infty}^{\infty} \left(\frac{y_s}{z} - \frac{y}{z^2}\right) e^{-r^2}\, e^{-y^2/z^2}\, \mu(zr)\, dr\, dz$$

Substituting now the Taylor's expansions (Eq. 5.36)for y and μ in the above equations we obtain that:

$$\mathcal{H}\mu \approx \frac{\mu_0\, \kappa(s)}{2} \sqrt{\pi\nu\delta t} + \mathcal{O}\Big((\nu\delta t)^{3/2}\Big)$$

So substituting the above result in the equation for the heat potential Eq. 5.33 we find:

$$\mu(s) \approx \mu_0(s) \approx -2\, F(s)\, (1 - \kappa(s)\sqrt{\pi\nu\delta t})^{-1} \tag{5.37}$$

Note that for the case of a cylinder of radius R, the curvature is constant ($\kappa = 1/R$) and for the case of a flat plate the curvature is zero so that the surface potential is only a function of the vorticity flux.

Substituting Eq. 5.37 in Eq. 5.35 we obtain an algorithm for updating the particle strengths in the domain so that the no-slip boundary condition is enforced.

5.4.5 A Test Case

We test our numerical algorithm for the enforcement of the no-slip boundary by simulating the flow induced by a cylinder oscillating about its axis with angular velocity $Q\sin(\sigma t)$. An analytic solution to this problem may be constructed by assuming the streamfunction of the flow to be of the form (Gray and Mathews, 1952):

$$\Psi \;=\; \phi(r)\, e^{i\sigma t}.$$

Under this assumption the vorticity field of the flow is given by :

$$\begin{aligned}\omega(r,t) \;&=\; A\cos(\sigma t)\,[kei_1(c)\,K^-(cr) \;-\; ker_1(c)\,K^+(cr)] \\ &-\; A\sin(\sigma t)\,[kei_1(c)\,K^+(cr) \;-\; ker_1(c)\,K^-(cr)]\end{aligned}$$

with the definitions

$$K^{\pm}(cr) \;=\; ker(cr) \;\pm\; kei(cr)$$

$$c \;=\; \sqrt{\frac{\sigma}{\nu}}, \qquad A \;=\; \frac{Q\,c}{\sqrt{2}}\,\frac{1}{ker_1^2(c) \;+\; kei_1^2(c)}$$

where $ker_1(x)$, $kei_1(x)$ and $ker(x)$, $kei(x)$ are the Kelvin's functions of order 1 and 0 respectively.

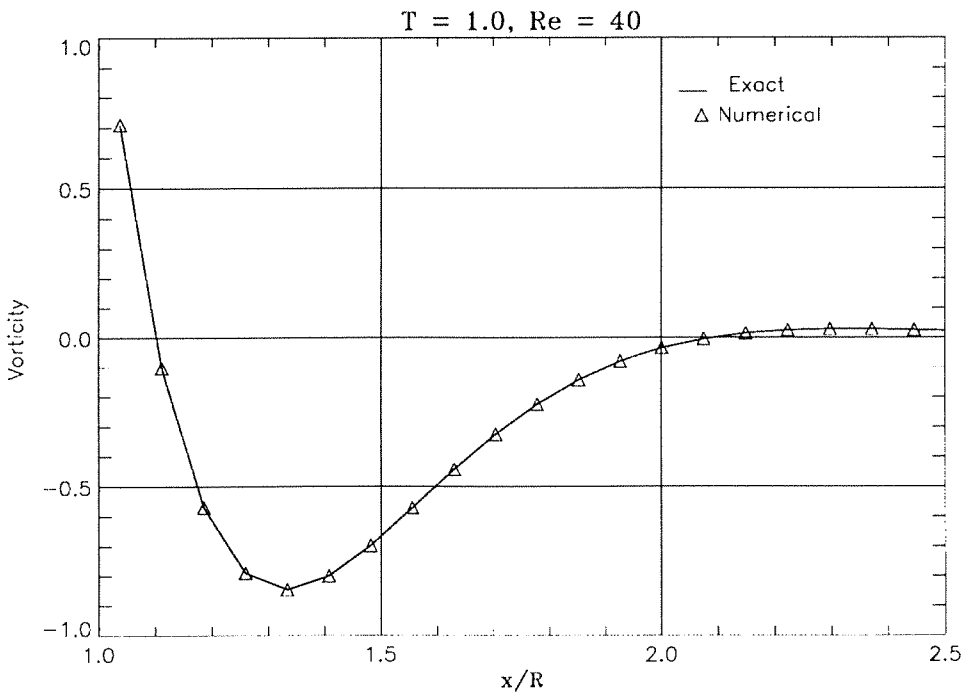


FIG. 5.6 Vorticity field around a purely rotating cylinder with $Q = 1$, $\sigma = 1$.

In Fig. 5.6 we show the results of the computed and analytical vorticity field for $T = 1$ and $\nu = 0.5$. In order to avoid the computation of the transient solution the initial vorticity field is computed from the analytic solution at the end of a period.

CHAPTER 6

Remeshing Schemes for Vortex Methods

In order for vortex methods to converge to solutions of the Navier-Stokes equations, the particles should overlap at all times. A computation is bound to become inaccurate once the particles cease to overlap. Computations involving non-overlapping finite core particles should be regarded then as modelling and not as direct numerical simulations. Excluding case specific initial particle distributions (e.g., particles placed on concentric rings to represent an azimuthally independent vorticity distribution) the loss of overlap (and excessive overlap) is an inherent problem of purely Lagrangian methods. The cause of the problem is the flow strain that clusters particles in one direction and spreads them in another in the neighborhood of hyperbolic points of the flow map, resulting in non-uniform distributions. At the onset of such particle distributions no error is usually manifested in the global quantities of the flow such as the linear and the angular momentum. However locally the vorticity field becomes distorted and spreading of the particles results in loss of naturally present vortical structures whereas particle clustering results in the appearance of unphysical ones, on the scale of the interparticle separation. The method is not able then to accurately resolve a smooth vorticity field. A characteristic example of this pathology is the simulation of the evolution of an axisymmetric Gaussian vortex using particles initially distributed on a rectangular grid. Although the linear and quadratic diagnostics of the flow do not exhibit any unphysical behaviour after almost two revolutions of the Gaussian an examination of the vorticity field reveals unphysical non-smooth vortical structures that eventually destroy the whole calculation.

This clustering and spreading of the particles has the following consequences for the present numerical scheme. For the *convection step*, it implies that a variable time step is necessary for the integration in different locations as it is desirable that particles do not travel more than one core radius at each time step For the *diffusion step*, the scheme of particle strength exchange imposes a further requirement on the particle distribution. In order to model diffusion properly each particle should be surrounded by others, so that it exchanges its circulation (see Ch.4) with them

according to the PSE scheme. These requirements imply an upper bound for the particle overlap equal to one. However, although one is tempted to use a value as close to one as possible, practice dictates that a computation with marginally overlapping particles would become inaccurate in shorter times. On the other hand a sensible value should be used for the lower bound to keep the number of particles to a minimum. Additionally, care should be exercised in the boundaries of the vortical regions so as to carry enough particles to accurately resolve the diffusion equation.

In the present numerical scheme we face the above issues with an adaptive computational strategy. The distorted (strained) net of vortices is mapped (*remeshed*) onto a uniform one by locally redistributing the strength of the vortex particles. This redistribution takes place once the particles cease to overlap at any location of the computational domain or they cluster in some region or there are not enough particles to properly resolve the diffusion step. To this effect the following algorithm is implemented:

A uniform grid of prescribed size is placed in the computational domain so as to cover the particles and extend far enough to capture diffusion effects in the boundaries of the vorticity distribution. The particle locations are examined with respect to the grid and we can identify the number of particles in each cell as well as examine if neighboring cells contain particles. Hence in cells where more than a certain number of particles exist this implies particle clustering and the vorticity field of its residents needs to be redistributed. On the other hand in regions where neighboring cells do not contain particles it is necessary to introduce new ones to ensure proper overlap and description of the diffusion scheme. This remeshing procedure is performed in $\mathcal{O}(N)$ operations and is vectorized.

6.1 Theoretical Considerations

The advantage of Lagrangian methods over classical Eulerian schemes is the adaptivity of the employed grid (the particles). However, a Lagrangian grid may become highly strained limiting the time to which a purely Lagrangian computation may be taken. We are interested then in replacing this (occasionally) strained grid with a new Lagrangian regular grid and simultaneously transport accurately the global flow quantities (such as the total circulation and impulse) from the old grid to the new. In the present scheme on the old particles we overlay a regular grid (the new particles) and the issue is how to interpolate the old vorticity field (ω) and (distorted) particle locations (x) to the new grid (particle) locations ($\tilde{x}$) and obtain a new vorticity field ($\tilde{\omega}$) such that :

$$\tilde{\omega}(\tilde{x}) \approx \omega(x)$$

or if $\tilde{\Gamma}, \Gamma$ denote the new and old particle strengths respectively then we are interested in determining an appropriate interpolation kernel Λ so that:

$$\tilde{\Gamma}_i(\tilde{x}_i) \approx \sum_{j=1}^{M} \Gamma_j(x_j)\, \Lambda(\tilde{x}_i - x_j)$$

The process is not of the usual interpolation type as it is complicated by the fact that the particles are disordered. The basic analysis of interpolation of this type is given by Schoenberg (1973). He has developed interpolation formulas that attempt to minimize the effect of the grid disorder on the interpolated quantity

6.1.1 Interpolation Kernels

We may distinguish between two types of interpolation formulae: the collocation type interpolation where $\tilde{\Gamma}(x_j) = \Gamma(x_j)$, and the smoothing interpolation where $\tilde{\Gamma}(x_j) \neq \Gamma(x_j)$. If h is the spacing of the new grid setting $u = |x|/h$ then interpolation kernels of the first type are:

Linear Interpolation

$$\Lambda_1(u) = \begin{cases} 1 - u, & \text{if } 0 \leq u \leq 1; \\ 0, & \text{otherwise.} \end{cases}$$

Second order Interpolation

$$\Lambda_2(u) = \begin{cases} 1 - u^2, & \text{if } 0 \leq u < 1/2; \\ (1 - u)(2 - u)/2, & \text{if } 1/2 \leq u < 3/2\ ; \\ 0, & \text{otherwise.} \end{cases}$$

Everett's formula of third order :

$$\Lambda_3(u) = \begin{cases} (1 - u^2)(2 - u)/2, & \text{if } 0 \leq u < 1; \\ (1 - u)(2 - u)(3 - u)/6, & \text{if } 1 \leq u < 2; \\ 0, & \text{otherwise.} \end{cases}$$

These kernels may be constructed in a systematic way (see below) so as to ensure conservation of as many moments of the vorticity field as possible. They do not have continuous derivatives and second and higher order kernels are not even continuous. This implies that when interpolating quantities having large fluctuations for small particle separation they might introduce large interpolation errors from small errors in the actual particle locations. The second type of interpolation attempts to minimize such effects. Schoenberg (1973) has introduced a set of such interpolation kernels the *central B-Splines*, (M_n). The formulae for the first four members of this family are given by :

$$M_1(u) = \begin{cases} 1, & \text{if } 0 \leq u \leq 1/2; \\ 0, & \text{otherwise.} \end{cases}$$

which is the nearest grid point(NGP) interpolation,

$$M_2(u) = \begin{cases} 1 - u, & \text{if } 0 \leq u \leq 1; \\ 0, & \text{otherwise.} \end{cases}$$

which is the linear interpolation ($\Lambda_1(u)$ and is usually the choice in CIC (Cloud In Cell) algorithms,

$$M_3(u) = \begin{cases} (3/4 - u^2)(2 - u)/2, & \text{if } 0 \leq u < 1/2; \\ (3/2 - u)^2/2, & \text{if } 1/2 \leq u < 3/2; \\ 0, & \text{otherwise.} \end{cases}$$

which is the first function to have a continuous first derivative and is usually referred to as the TSC (Triangular Shaped Cloud) function, and finally :

$$M_4(u) = \begin{cases} 2/3 - u^2 + 1/2, & \text{if } 0 \leq u < 1; \\ (2 - u)^3/6, & \text{if } 1 \leq u < 2; \\ 0, & \text{otherwise.} \end{cases}$$

Note that with increasing order the interpolation is less sensitive to the disorder of the particles as information is sampled from a larger area.

The analysis of those schemes is further facilitated by considering their form in the transformed Fourier space. Returning to the realm of vortex methods and for simplicity to the one dimensional case we consider a particle of unit strength having core radius σ located at (x_0) having a shape-function $\phi((x-x_0)/\sigma)$. We are interested in replacing this particle by M particles at equally spaced locations $(\tilde{x}_m)$, having strength $(\tilde{\Gamma}_m)$ so as to obtain an 'equivalent' vorticity distribution. The vorticity at a location x is given as :

$$\begin{aligned} \omega(x) &= \phi((x-x_0)/\sigma) \\ &\approx \sum_{m=1}^{M} \tilde{\Gamma}_m \, \phi((x-\tilde{x}_m)/\sigma) \;=\; \tilde{\omega}(x) \end{aligned}$$

(Note that ϕ should not be confused with the interpolation kernels.) It is obvious that one can never expect an 'exact' solution to the above equation for the unknown values $\tilde{\Gamma}_m$, for all locations x. However one may try to :

- Conserve as many moments(T_a) of the vorticity , $(T_a \;=\; \int \omega(x)\, x^a \, dx)$, as possible $(a \;=\; 0,1,2,...)$ and
- Minimize the error at distances larger than the mesh size.

Taking the Fourier Transform of the previous equation we have :

$$\begin{aligned} \mathcal{FT}\{\,\omega(x)\,\} &= e^{-ikx_0}\, \mathcal{FT}\Big\{\,\phi(\frac{x}{\sigma})\,\Big\} \\ &\approx \mathcal{FT}\Big\{\,\phi(\frac{x}{\sigma})\,\Big\} \sum_{m=1}^{M} \Gamma_m \, e^{-ikx_m} \end{aligned}$$

or equivalently we ask that the new particles have to satisfy the following equation:

$$\sum_{m=1}^{M} \tilde{\Gamma}_m \, e^{ik(x_0-\tilde{x}_m)} \;\approx\; 1$$

Defining $\Delta_{0m} \;=\; x_0 - \tilde{x}_m$, and expanding the left hand side for small values of $\tau_{0m} \;=\; k\Delta_{0m}$ (as the smoothing kernels decreases exponentially with k) we obtain:

$$\sum_{m=1}^{M} \tilde{\Gamma}_m \,(1 \;+\; i\tau_{0m} \;-\; \frac{1}{2}{\tau_{0m}}^2 \;-\; \mathcal{O}({\tau_{0m}}^3)) \;\approx\; 1$$

Having M as the free parameter in the above equation we see that we can approximate the vorticity field to the extent possible by requesting that M terms are satisfied in the above equation. Note that these equations are independent of the shape function ϕ and the core radius σ. We may systematically construct then interpolation kernels that obey the above requirements by employing more and more mesh-points. However computational cost becomes a factor and one usually avoids using $M > 4$. For $M = 1$ we can conserve the circulation only by assigning the circulation of each particle to its nearest mesh point (the NGP scheme). Solving the above sets of equations for $M = 2, 3, 4$ we obtain respectively the interpolating kernels $\Lambda_1, \Lambda_2, \Lambda_3$, which conserve respectively the first, second and third moment of the vorticity field by using $2^d, 3^d, 4^d$ mesh points (with d the dimension of the problem). Those interpolation schemes act then as low pass filters removing all the high order harmonics that arise from the irregularity of the grid. Simultaneously they determine the smaller scales of the vorticity field we expect to resolve in our simulations.

The B-Splines act as better low pass filters (their Fourier transform is given by $(\sin(\pi kh)/\pi kh)^n$), however they conserve only the circulation and the linear momentum of the vorticity field. B-Splines have an accuracy of $\mathcal{O}(h^2)$ so they can interpolate exactly only linear functions. Monaghan (1985) presents a method to improve their accuracy while maintaining their smoothness. In general B-Splines are able to accurately interpolate smooth functions but they fail to represent functions with steep gradients. Higher order harmonics introduced by the particle disorder are dampen at the expense of additional numerical dissipation.

An extensive survey of the Fourier space analysis of these schemes may be found in the book of Hockney and Eastwood (1981).

6.2 Practical Considerations

6.2.1 Choice of interpolation kernel

In the present simulations the Λ type kernels were employed. One reason is that in the present scheme a regular Lagrangian grid is restored every few time steps and so that the interpolation does not occur from a highly disordered set of points. Moreover their ability to interpolate functions with steep gradients make them more attractive when one needs to consider remeshing in the presence of a solid boundary (see 6.2.4). *B-splines* seem more appropriate for simulations where the Lagrangian grid is not regularized every few time steps but is continuously employed and we need to interpolate flow quantities onto a regular grid (like in CIC and SPH schemes). The smoothness of these functions helps in minimizing the error arising from a highly distorted grid, however at the expense of significant numerical dissipation.

6.2.2 Vectorization

Although the remeshing procedure does not take place every time step it is important for a fast code to have a vectorized procedure. This issue is even more important in PIC codes where this particle-grid interpolation is an important component of the algorithm (Haddard et al. (1991)) In the present code vectorization is achieved by a two-step procedure. First we map the locations of old and the new (candidates) particles onto an index array. The active indices of the array are determined from the old particles and subsequently the mapping is reversed to obtain the new particle locations. At the expense of additional storage (two sets of particle locations and strengths plus the index array) this procedure avoids overwriting and permits an efficient vectorization on vector machines with gather-scatter capabilities such as the CRAY/YMP. The remeshing algorithm scales as $\mathcal{O}(N)$ and is fully vectorized.

6.2.3 Remeshing in an Unbounded Domain

In the present simulations we chose to conserve the circulation, linear and angular momentum of the vorticity field In a two dimensional domain this would require 6 mesh points for each particle and such schemes may be easily devised (Hockney and Eastwood, 1981). However in order to maintain the properties of the one dimensional schemes and avoid the complicated programming , two dimensional schemes which are cartesian products of their one dimensional counterparts are employed. The interpolation kernel is defined then as:

$$\Lambda(x,y) \;=\; \Lambda(x)\Lambda(y)$$

Using Λ_2 each particle located in a cell (I,J), affects 9 mesh points presented as shaded areas in Fg.3.1. Here a rectangular grid was used to re-initialize the particles but similar schemes may be devised for other types of grids as well.

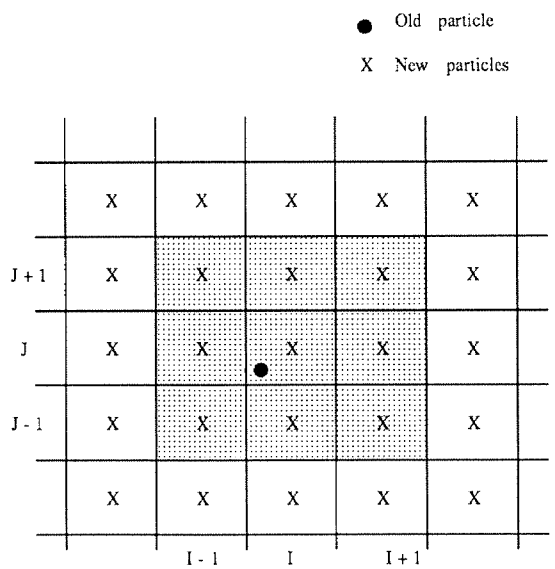


FIG. 6.1 Detail of the re-initialization grid. The shaded cells are affected by the pre-remeshed particle, denoted by a bullet in the sketch

6.2.4 Remeshing in a Bounded Domain

When boundaries are present the remeshing procedure is complicated as the new mesh-points have to be outside the body. It is obvious then that the schemes employed for an unbounded domain have to be modified for particles that are located in an (I,J) cell which is adjacent to the boundary. Here we devise a scheme to account for this difficulty. It requires again 9 points and conserves the same quantities as for the unbounded case. In Fig. 6.2 the nine cells affected by a cell adjacent to the wall are depicted. The interpolating kernel is again the product of two one dimensional forms but now

$$\Lambda(x,y) \;=\; \Lambda_I(x)\Lambda_J(y)$$

with $\Lambda_I \;=\; \Lambda_2$ and

$$\Lambda_J \;=\; \begin{cases} 1 \;-3/2\; v \;+1/2\; v^2, & \text{for cells } I; \\ v\,(2 \;-\; v), & \text{for cells } I+1; \\ v\,(v \;-\; 1)/2, & \text{for cells } I+2; \\ 0, & \text{for all other } I \end{cases}$$

where $v \;=\; (x \;-x')/h$ and I here denotes the off-boundary direction.

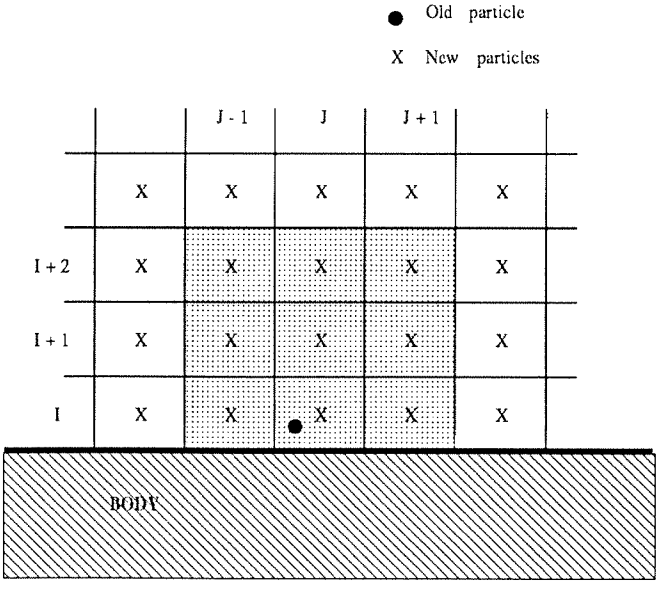


FIG. 6.2 Detail of the re-initialization grid adjacent to a wall. The shaded cells are affected by a particle adjacent to the wall, denoted by a bullet in the sketch. Note the region of influence compared to that in Fig.6.1

6.2.5 Time Stepping After Remeshing

In our simulations an Adams-Bashforth scheme was used to integrate the trajectories of the particles. This requires at each time level (t_n) the knowledge of the velocity field on the particle at instants (t_{n-1}) and (t_n). However as the Lagrangian grid is restored every few time steps the velocities of the newly created particles are not readily available. One may avoid to face the issue by integrating using an Euler time step immediately after remeshing (if we wish to avoid the complications of the present algorithm when using a multi-step scheme such as the Runge-Kutta). An Adams-Bashforth scheme may still be used however if along with the particle strengths we interpolate the velocity field (U) of the particle as well. For the newly restored Lagrangian particles at time level (t_n) we would have that :

$$\tilde{U}_i(\tilde{x}_i\ , t_{n-1}) \ = \ \sum_{j=1}^{M} U_j(x_j\ , t_{n-1})\ \Lambda(\tilde{x}_i \ - \ x_j)$$

The use of such a scheme has the drawback that the relation $\omega \ = \ \nabla \times U$ is violated for the remeshed vorticity field.

Alternatively then the following two-step procedure may be implemented: At time level (t_{n-1}) the particles are mapped onto a regular grid and the velocity field is computed on these mesh points $(\tilde{x})$ via the Biot-Savart law. The velocity field on the particle locations $(x(t_{n-1}))$ is computed as well and the particles are advanced according to the AB scheme to new locations $(x(t_n))$. There again the particles are mapped onto the same mesh points which become now the new Lagrangian particles for our simulation. They may be advanced then using an AB scheme using the velocity field computed in the previous stage. This procedure although it is more time consuming and results in more complicated programming it helps to maintain the second order accuracy in the time integration throughout the simulations.

6.3 An Alternative Remeshing Scheme

An alternative technique for the rezoning of the distorted grid associated with the particles may be devised based on a technique initially applied to the Arbitrary Lagrangian-Eulerian (ALE) method. In this interpolation the regridding procedure may be carried out using an integral formulation for the remapping of the vorticity field which is :

$$\tilde{\Gamma}_i = \int\int_{Ai} \omega(\vec{x})\, d\vec{x}$$

where $\tilde{\Gamma}_i$ is the circulation assigned to the particle in cell A_i of the new mesh and $\omega(\vec{x})$ is the vorticity of the old (distorted) mesh. The above integral may be expressed in discrete form as

$$\tilde{\Gamma}_i = \sum_j \Gamma_j \frac{A_{ij}}{A_j}$$

where A_{ij} denotes the overlapping area of the old cell i and the new cell j (the shaded area in Fig.3.3). This remeshing procedure involves then the computation of those overlap areas. An efficient algorithm for a piecewise constant vorticity field is presented in Dukowicz and Kodis (1987).

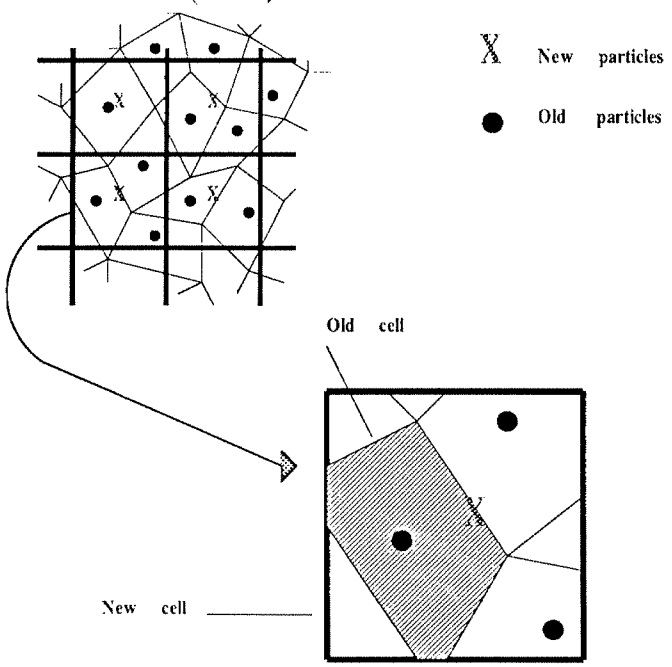


FIG. 6.3 Reinitialization of a distorted Lagrangian grid. The shaded region denotes the fraction of the circulation associated with the particle of the old cell contributing to the circulation of the particle in the new cell.

Using the identity $\nabla^2\Psi = -\omega$ the surface integral is translated into a line integral

$$\tilde{\Gamma}_i = \oint_{C_k} \vec{u} \cdot d\vec{l}$$

along the contour C_k that defines the overlap region of the old and the new cell. This integral may be efficiently computed by traversing first the cells of the old grid and then the cells of the new grid and accordingly add the contributions.

In comparison with the method of choice described in the previous section the ALE technique is more complicated , hard to vectorize and it introduces additional numerical dissipation as it conserves only the circulation of the flow. However it seems advantageous when one is dealing with remeshing of a vortex field around a complex configuration. There the interpolations described before encounter problems near the boundaries as they have to be modified for each geometry. The ALE method does not care about the specifics of the boundary as it deals directly with the particle cells that may have any geometric configuration.

6.3.1 Summary

In conclusion the present remeshing procedure allows for the progressive inclusion of particles in the computation that are activated by the diffusion process. Such particles would have to exist a priori to account for proper diffusion but that would burden the simulation with extra particles that are activated only at a few time steps. The procedure also maintains a minimum level of resolution at all stages of the computation in the regions of vorticity.

CHAPTER 7
Simulations of Vortical Flows

In this chapter we discuss the application of the method described in the previous chapters to the simulation of unsteady vortical flows. First we discuss the application of vortex methods to the infinite domain. More specifically we consider the inviscid evolution of a particular elliptical (2:1) vorticity distribution. Then we focus our attention to the simulation of unsteady separated flows behind a circular cylinder.

7.1 Inviscid Evolution of a 2:1 Elliptical Vortex

As purely Lagrangian methods for the simulation of vortical incompressible flows we may characterize the point vortex method (Rosenhead, 1931) and the method of contour dynamics (Zabusky et al., 1979). Point vortex methods rely on the pointwise approximation of the vorticity field whereas contour dynamics resolve regions of piecewise constant vorticity by tracking the evolution of its boundary. These methods are inherently inviscid thus providing a superior alternative to Eulerian schemes for the simulation of inviscid flows. However inviscid flows are distinguished by the wide range of scales they develop and they require an increasing number of computational elements with the progression of time to resolve them. Moreover Lagrangian methods are faced with the challenge of grid distortion and the formation of unphysical structures.

Whereas vortex methods are faced with the problem of particle clustering and scattering (Ch.6), contour dynamics are faced with the problem of singularity formations on the contours ('wave breaking', etc.). The art of removing such singularities and introducing more computational elements when deemed necessary consists the method of 'contour surgery' (Dritschel, 1989). A series of computations have been performed (such as vortex merging, filamentation, vortex equilibria etc.), using these methods, for a variety of intial conditions and the interested reader is referred to the review article of Pullin (1992) for further details.

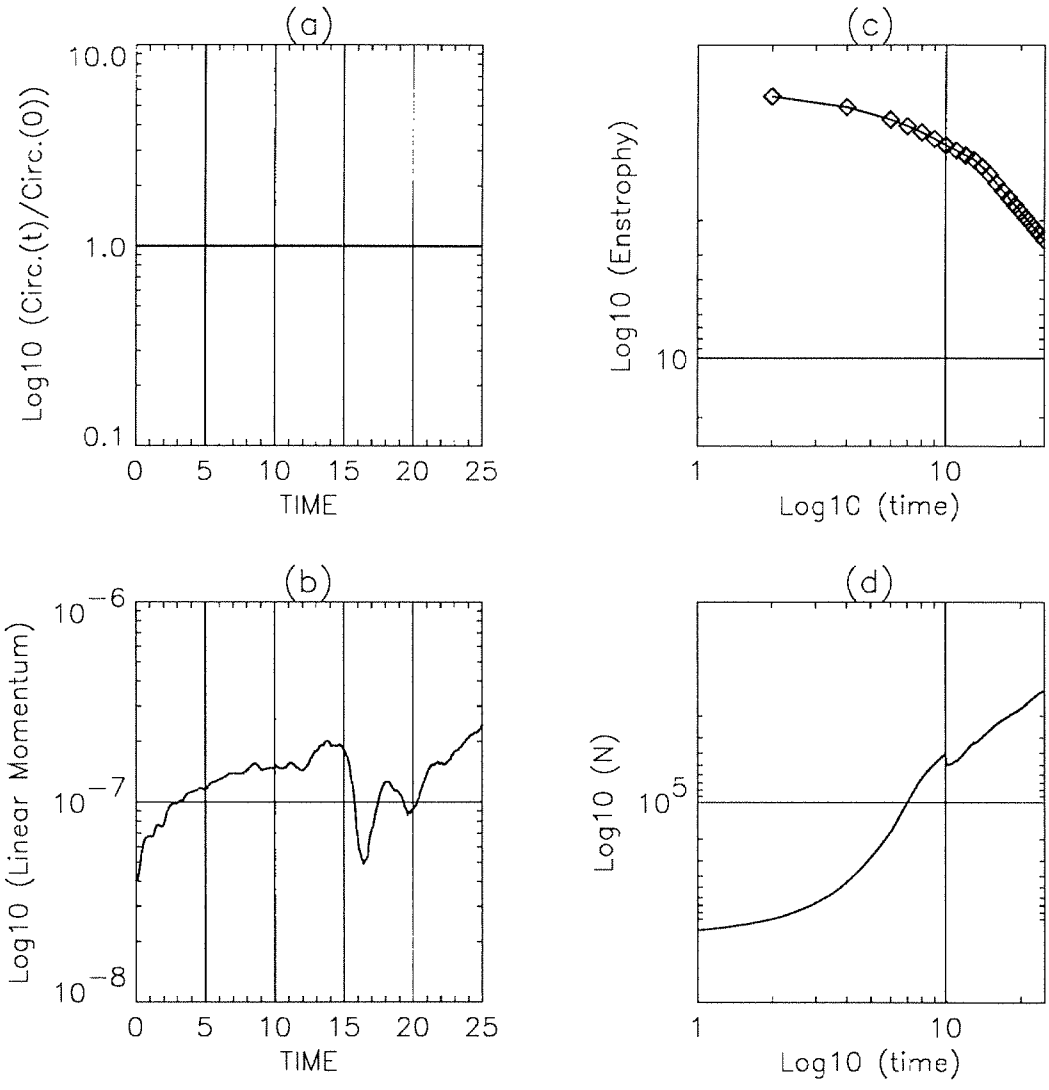


FIG. 7.1 Diagnostics for the inviscid evolution of a 2:1 elliptical vortex

Our scope is to demonstrate that, in their present implementation, vortex methods offer a viable alternative for Lagrangian simulations of the viscous/inviscid evolution of arbitrary continuous vorticity distributions. To demonstrate the validity of our approach, for simulations over extended times, we consider the evolution and filamentation of an elliptical (2:1) vorticity distribution. The initial vorticity profile is determined by:

$$\omega(r,\phi) \;=\; \Lambda \begin{cases} 1 \;-\; f_q(r(\phi)/R_0(\phi)), & \text{if } r \le R_0, \\ 0, & \text{otherwise.} \end{cases}$$

where:

$$f_q(x) \;=\; \mathrm{e}^{-\frac{q}{x}\,\mathrm{e}^{\frac{1}{x-1}}}$$

In the present simulations the values $q = 2.56085$ (so that $f_q = 0.5$) and $\Lambda = 20/\pi$ and $R_0 = 0.9$ were selected. This idealized initial profile was suggested by Melander et al. (1987) and it is characterized by steep vorticity gradients as about 90% of the vorticity variation occurs in the interval [0.20, 0.70]. For our computations a core radius $\epsilon^2 = 2.5\,10^{-5}$, an overlap ratio of 0.90 and a cutoff circulation limit between 10^{-7} and 10^{-5} were selected. During the course of the computation we varied several parameters of the computational scheme (such as time step, remeshing intervals, number of the multipole expansions etc.) to examine the robustness and detect problems of our numerical code. A time series evolution of this configuration is presented in Fig. 7.2 and Fig. 7.3. One may observe the progressive filamentation of the ellipse and the appearance of smaller scales in our computations (Fig. 7.4).

In Fig. 7.1 diagnostics of the simulation are presented. One may observe the conservation of the circulation and the linear momentum of the flow demonstrating the accuracy of our simulations. On the other hand one may observe the appearance of asymmetries in our computations. Such asymmetries should be attributed to the reduced number of multipole expansions used during some intervals of the present simulation. As more complex structures appear, due to the evolution of the ellipse, our code automatically generates (through remeshing) new particles to resolve such regions. Initially two major filaments appear due to the differential rotation of the ellipse that subsequently reattach to the ellipse and extract further vorticity. The interplay of these filaments with the main vorticity distribution is reduced with the progression of time as the filaments become elongated and their local strength is reduced. At the end of the simulation (corresponding to about three rotations) we observe that the main elliptical distribution continuous to evolve surrounded by a fair number of passively advected, weak vorticity structures (filaments). The present computations present no evidence of axisymmetrization. The ellipse continues to shed filaments and its shape oscillates without ever attaining a circular stable configuration. Although the evidence is such that our simulations are accurate we should not try to conclude more than what the exploratory nature of these computational experiments allows. A more systematic study of the evolution of the elliptical configuration is in order with carefully chosen initial conditions so as to compare the results of the present method with results of the method of contour surgery (Dritschel, 1989) and the pseudo-spectral simulations of Melander et al (1987).

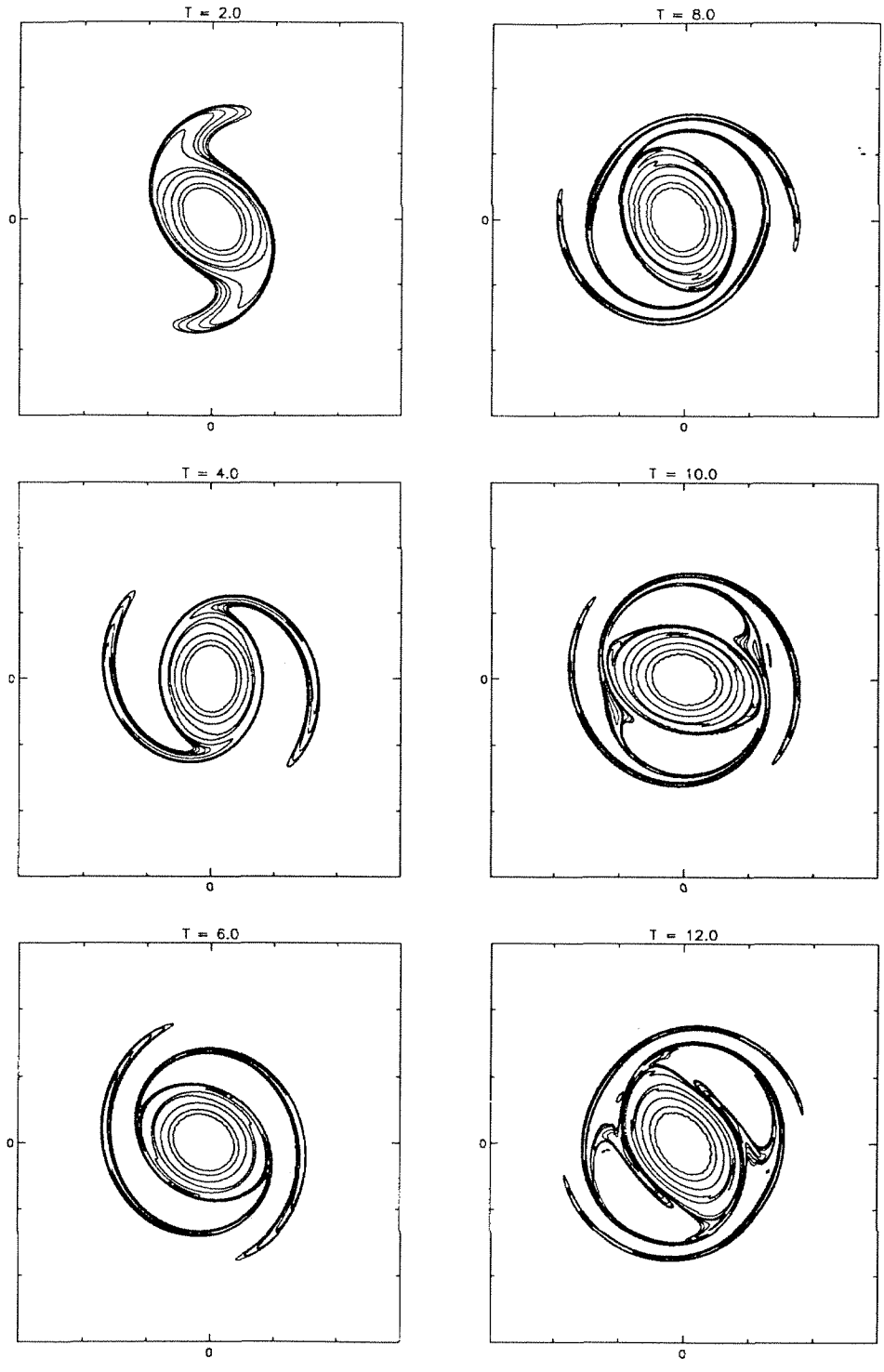


FIG. 7.2 Time frames for the inviscid evolution of a 2:1 elliptical vortex

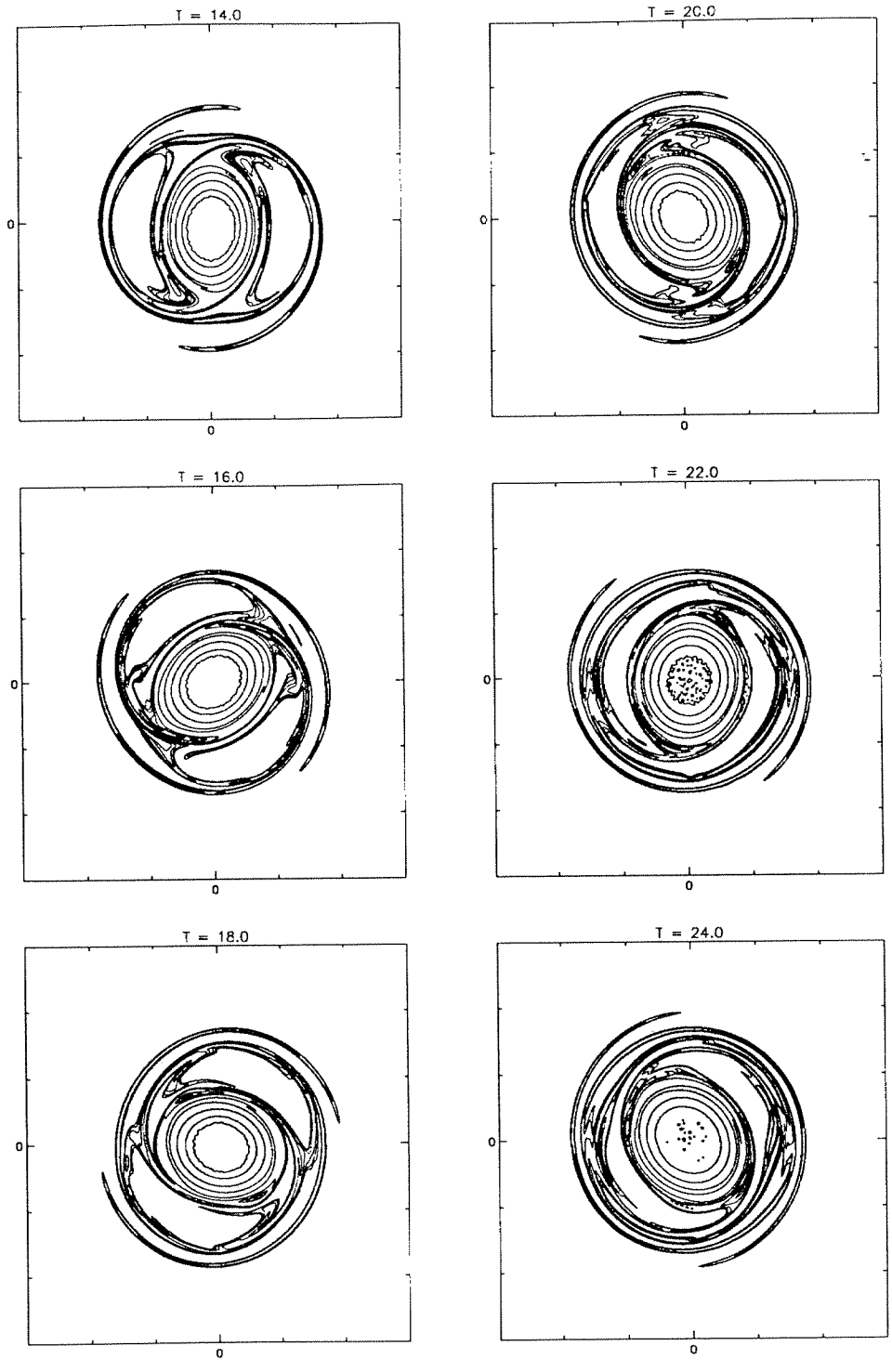


FIG. 7.3 Time frames for the inviscid evolution of a 2:1 elliptical vortex

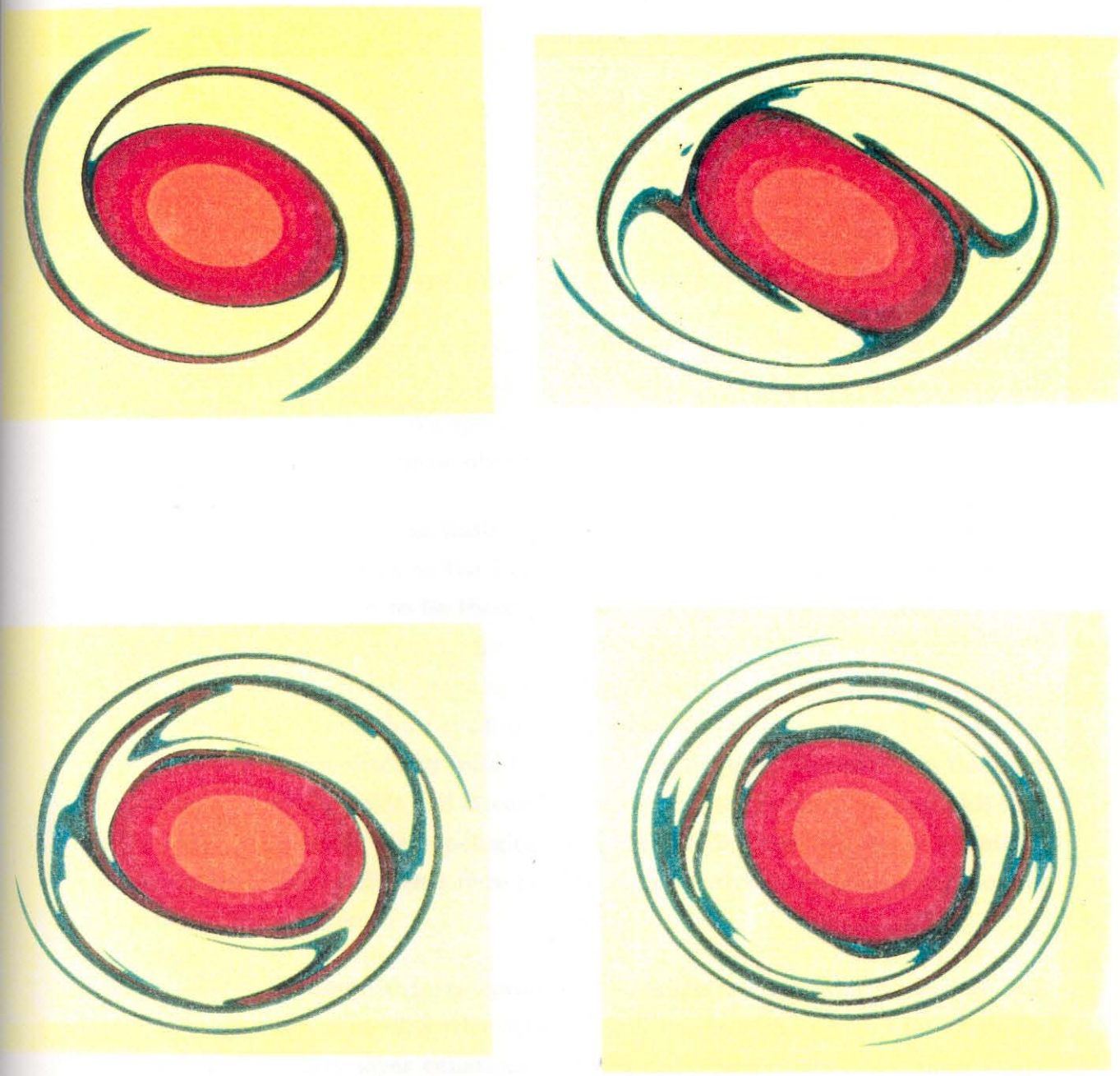


FIG. 7.4 Vorticity field of an elliptical (2:1) vortex. Frames are from left to right and top to bottom (T= 6.0, T = 12.0, T = 18.0, T = 24.0).

7.2 Simulations of Unsteady Flows Past a Circular Cylinder

This section discusses the application of the vortex method described in previous chapters to unsteady bluff body flows. The flow behind a circular cylinder started impulsively in rotation and/or translation is investigated as a prototype of unsteady separated flows. Computations are presented for Re = 40 to 9500 in order to validate our numerical method and gain some insight into the physical mechanisms present in such flows.

7.2.1 Brief Review of Experimental and Computational Results

Here we review some of the experimental and computational studies for the early stages of the development of the wake of an impulsively started translating and or rotating circular cylinder. We specifically discuss the investigations the results of which will be compared to those obtained by the present scheme.

Experiments : Works on unsteady flows resulting from an impulsive acceleration of a cylinder date back to the Prandtl era (1925). However the most extensive experiments to date seem to be those presented by Bouard and Coutanceau (1980) (referred to as *BC* from here on) for the translation only case and Coutanceau and Menard (1985) ,(*CM*), for the rotating and translating cylinder. They used a specially designed apparatus to produce quasi-instantaneous starts and analyzed photographically the flow field visualized by solid tracers uniformly put in suspension in the fluid. Instantaneous velocity fields and streamlines are presented in their works which they use to analyze in detail the topological structure of the flow. However quantities such as drag and lift coefficients were hard to obtain with this technique and are not presented in their papers.

Analysis : Theoretical investigations of an impulsively started flow were first undertaken by Blasius in 1908 who obtained the first two terms of a time series solution of the boundary layer equations. Subsequently there have been many works attempting to obtain higher order terms and advance the solution beyond the separation stage but the most notable ones are those of Collins and Dennis (1973) and Bar-Lev and Yang (1975) (referred to as *CD* and *BY* respectively from here on). In *CD* they formulate the problem in boundary layer variables and obtain expansions in powers of time. These expansions are corrected to account for finite Re effects

and they are adjusted to match the uniform flow far from the cylinder. In BY the vorticity equation is solved by the method of matched asymptotic expansions. Inner (rotational flow) and outer (potential flow) solutions are obtained to third order in time and a composite solution is formed. Both works provide extensive information for flow quantities of interest (such as vorticity field, streamlines, body forces) that are valid for short times. Their region of validity increases with increasing Re. They are the yardstick which many investigators use to test their numerical scheme at least for the initial stage of the flow.

Computations : Impulsively started flows present a serious challenge for the numerical method. Difficulties exist in the formulation of the boundary and initial conditions of the problem. High resolution simulations are necessary for high Reynolds numbers to adequately resolve the singular (due to the impulsive start) character of the flow.

The first computations were presented in the late 50's by Payne (1958) for relatively low Re numbers (40 to 100). Numerous computations have been performed in the last 30 years on this flow but there are still open questions as to whether the numerics do not overwhelm the physics of the problem especially for high Re flows. The problem is usually formulated in vorticity - streamfunction variables and Eulerian, Lagrangian and hybrid methods have been used for their discretization.

Ta Phuoc Loc (1980) (TL from here on) uses a fourth order scheme to resolve Poisson's equation for the streamfunction and a second order finite difference scheme for the vorticity transport equation. He presented computations for a range of Re numbers (550 to 9500) and detailed diagnostics and comparisons with experimental results (see also Ta Phuoc Loc and Bouard, (1985)). Lecointe and Piquet (1985) have tested several high order compact finite difference schemes as well and they present accurate computations for Re up to 550 and tentative simulations for higher values of Re. Their diagnostics however are limited and not conclusive. A more recent work is that of Wang and Dalton (1991). They use a predictor - corrector finite difference scheme for the vorticity transport equation and a Fast Poisson solver for the streamfunction. They present results for impulsively started and stopped flows for Re of 102 and 550.

In vortex methods the most notable works are those of Smith and Stansby (1988) (SS) and the more recent ones of Chang and Chern (CC). Both schemes use the

method of CIC (Christiansen . 1973) to convect the vortices but use different techniques to account for viscous effects. *SS* use the method of random walk whereas *CC* use a finite difference scheme on the grid used by the CIC technique to resolve the diffusion equation. Both works take advantage of the stability properties of vortex methods to extend their computations to very high Reynolds numbers (up to 10^6 in *CC*). However it appears that the increase in the Re simulated is not followed by an adequate increase in the resolution.

Also *CC* present results for the case of an impulsively started rotating circular cylinder. By far however the most extensive and accurate results on this flow are those presented in Badr and Dennis (1985) and the combined experimental and computational work of Badr et al. (1990) for Re up to 10^4. They use Fourier analysis to reduce the governing partial differential equations to a set of time dependent equations in one space variable. By truncating these Fourier series to a finite number of terms a set of differential equations is obtained that is used to compute the flow.

The above mentioned experimental, theoretical and computational investigations are not at all a complete list of the available literature. They are however representative works with which we compare our numerical scheme to obtain an understanding of its validity and capabilities as it is discussed in the following sections.

7.2.2 A Note on Impulsively Started Flows

One may recognize that there cannot be experimentally such a thing as a truly impulsive start, although sufficiently *rapid* starts may be considered as producing a quasi - instantaneous motion of the body.

In theory an impulsive start may be formulated using the potential flow field as the initial condition. At time $t = 0^+$ a potential flow exists and a slip velocity (vortex sheet of zero thickness) is observed on the surface of the body.

Numerical schemes encounter difficulties with impulsive starts as well as it is difficult to resolve the initially developed thin boundary layers. Flow features such as streamline patterns are fairly well described by several numerical schemes and are in good agreement with experimental visualization. However most schemes encounter difficulties in accurately describing the vorticity field and calculating quantities such as the drag coefficient that exhibits a $1/\sqrt{t}$ singularity.

7.2.3 Diagnostics

The Reynolds number (Re) of the flow is defined based on the diameter of the cylinder (D), as :

$$Re = \frac{\mathrm{U\,D}}{\nu}$$

and the time (t) is nondimensionalized based on the radius (R = D/2) of the cylinder as :

$$T = \frac{\mathrm{U\,t}}{\mathrm{R}}$$

The angular velocity (Ω) of the cylinder is nondimensionalized by :

$$\alpha = \frac{\Omega \mathrm{R}}{\mathrm{U}}$$

The principal variable of our scheme is the vorticity field which is described as the linear superposition of the vorticity field of the individual particles. One may compute diagnostics such as the streamlines, body forces and velocity field using the strength and locations of those vortices.

Body Forces : The net force exerted on the body from the fluid (of unit density) may be expressed as :

$$\mathbf{F_b} = -\frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{fluid}} \mathbf{u}\, d\mathbf{x}$$

In order to carry out the integration over the whole domain, covering both the body and the fluid we add and subtract the term $\frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{body}} \mathbf{u} d\mathbf{x}$ so that :

$$\mathbf{F_b} = -\frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{fluid}+\text{body}} \mathbf{u}\, d\mathbf{x} + \frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{body}} \mathbf{u}\, d\mathbf{x}$$

The velocity field inside the body is defined by:

$$\mathbf{u}|_{\text{body}} = \mathbf{U_b} + \Omega\, \hat{\mathbf{e}_z} \times (\mathbf{x} - \mathbf{x}_b)$$

where $\mathbf{x}_b$ denotes the center of mass of the body and $\mathbf{U_b} = d\mathbf{x}_b/dt$. By extending the definition of the vorticity field inside the body (assuming a constant vorticity distribution with strength $\omega_b = 2\Omega$) we may write for the force on the body that:

$$\mathbf{F_b} = -\frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{fluid}+\text{body}} (\boldsymbol{\omega} \times \mathbf{x})\, d\mathbf{x} + \frac{\mathrm{d}}{\mathrm{d}t} \int_{\text{body}} [\mathbf{U_b} + \Omega\, \hat{\mathbf{e}_z} \times (\mathbf{x} - \mathbf{x}_b)]\, d\mathbf{x}$$

which reduces to :

$$\mathbf{F_b} = -\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{\text{fluid}+\text{body}} (\boldsymbol{\omega} \times \mathbf{x})\, d\mathbf{x} + A_B \frac{\mathrm{d}\mathbf{U_b}}{\mathrm{d}t}$$

where A_B denotes the area of the body. Furthermore the integral involving the vorticity over the area of the body may be computed explicitly so that a final expression may be obtained for the force on the body :

$$\mathbf{F_b} = -\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{\text{fluid}} (\boldsymbol{\omega} \times \mathbf{x})\, d\mathbf{x} + A_B \frac{\mathrm{d}\mathbf{U_b}}{\mathrm{d}t} - 2\, A_B \frac{\mathrm{d}}{\mathrm{d}t} \left(\Omega\, \widehat{\mathbf{e}_z} \times \mathbf{x}_b\right) \qquad (7.1)$$

The vorticity in the wake is described in our method by the superposition of the vorticity field of the individual particles so the integral over the domain occupied by the fluid may be computed in our method by: to :

$$\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{\text{fluid}} \boldsymbol{\omega} \times \mathbf{x}\, d\mathbf{x} = \frac{\mathrm{d}}{\mathrm{d}t} \sum_{i=1}^{N} \Gamma_i\, \mathbf{x}_i \times \widehat{\mathbf{e}_z}$$

where $\mathbf{x}_i = (x_i, y_i)$ is the location and Γ_i the strength of the vortices in the wake.

The drag and lift coefficients (c_D and c_L respectively) of the body are given then by:

$$c_D = \frac{2\mathbf{F_b} \cdot \widehat{\mathbf{e}_x}}{U_\infty^2\, D}$$

$$c_L = \frac{2\mathbf{F_b} \cdot \widehat{\mathbf{e}_y}}{U_\infty^2\, D}$$

where D is the characteristic length of the body.

A note should be added here regarding the above formula for the computation of the drag coefficient. As was mentioned in Ch.6 a remeshing step is periodically performed as the particles become distorted and the simulations fail to be accurate. In this remeshing step the linear impulse is conserved however its time derivative (the drag) is not necessarily conserved. One must be careful then how the time derivative of the linear impulse is discretized. A reasonable approach is to use a backward difference scheme to compute the drag at the time of remeshing, a forward scheme after remeshing and a central differencing scheme at all other time steps. This approach avoids differentiating across remeshing steps thus avoiding the problem induced by remeshing. Alternatively a Box - Carr type smoothing may be applied

to the linear momentum. The smoothed impulse may be subsequently differentiated in obtain the drag coefficient. Both approaches have been used and they result in almost identical results. However the effects of remeshing are more prominent at higher Re as it is performed more often and this process produces the small glitches observed in the drag curves.

Streamlines : The streamlines of the flow may be computed as a linear superposition of the streamlines induced by the individual particles.

$$\Psi(\mathbf{x}_m) \;=\; -\frac{1}{2\pi}\sum_{i=1}^{N}\Gamma_i \log|\mathbf{x}_m \;-\; \mathbf{x}_i|^2 \tag{7.2}$$

Note that the point vortex streamfunction is used. This approach is dictated by the presence of the body. The use of smooth vortices near its surface would imply a non constant streamfunction inside the body thus violating the impermeability condition. In order to compute the streamfunction on a grid a fast algorithm is implemented similar to the one presented in Ch.3. At each grid point the influence of the nearby particles is computed using Eq. 7.2 and multipole expansions are used to compute the influence of particles sufficiently removed from that point. This helps reducing the cost from $\mathcal{O}$(MN) to $\mathcal{O}$(N + MlogN) where M is the number of grid points on which the streamfunction is computed.

Body Vorticity : The computation of the vorticity field on the body has been traditionally a puzzling issue for vortex methods. Here we use a technique borrowed from the finite difference community. The vorticity on the body is given by :

$$\omega_{\text{body}} \;=\; -\nabla^2\Psi$$

As the streamfunction may be computed on a grid around the body a finite difference operator may be applied to the Laplacian so that the vorticity is computed as (Gresho, 1992) :

$$\omega_{\text{body}} \;=\; (7\Psi_0 \;-\; 8\Psi_1 \;+\Psi_2)/2h^2$$

where, if $y = 0$ describes locally the body surface, Ψ_0 is the value of Ψ on the body and Ψ_1, Ψ_2 the values at grid locations $y = h$, $y = 2h$ respectively. Note that the spacing h for the computation of the streamfunction should be larger than the smoothing core of the particles as the point vortex formula is used for the computation of the streamfunction.

Velocity Field : The velocity field is computed using the Biot Savart law. The smooth Biot-Savart kernel is used to avoid spurious results at locations near the centers of the vortices. Our fast algorithm for the convection step is used herein to compute efficiently velocities at arbitrary locations in the domain.

7.2.4 Results

As discussed in previous sections the flow behind an impulsively started cylinder has been the subject of a large number of experimental, theoretical and computational works. Experiments can be truly two - dimensional for the early stages of the flow providing results with which theoretical and computational works may test their validity and approach.

7.2.4.1 The Impulsively Started Translating Cylinder

The key features of this flow are extensively discussed in the experimental work of Bouard and Coutanceau (1980). The flow is known to remain symmetric, at least for small times, for Re up to 9500. In our computations **no symmetry constraint** is imposed. Perturbations introduced by roundoff errors could trigger an asymmetric flow however. Our computations follow the experimental trends as the flow remains symmetric for longer times with lower Reynolds numbers.

In Fig. 7.5 and Fig. 7.6 the drag coefficient as computed from the present method is compared with the theoretical values of BY and CD. At the onset of the flow the drag exhibits a square root singularity, a challenging indeed behaviour for numerical schemes. The accuracy of the present vortex method is exhibited as it is able to capture this trend for all values of Re. It is interesting to observe that for all computations a remarkable agreement is found for times up to $T \approx 0.20$ beyond which the theoretical and computational results diverge. Note also that a discrepancy appears in the two theoretical models for a wide range of Re. It is at this value of T that the drag coefficient starts to recover from the sudden drop due to the impulsive start and the effects of convection become important.

- **Re = 40**

Vortex methods have been used in the past as a tool for the simulation of large Re number flows producing questionable results when applied to flows of large viscosity. This is mainly attributed to the problem vortex methods had in dealing with diffusion and the no-slip boundary condition. This difficulty is alleviated with the present approach and highly viscous simulations can be indeed handled.

At Re = 40 the flow is known to reach a steady state. In the present scheme the wake is still continuing to grow at the end of the simulation. Our goal was to terminate the computation when the drag coefficient (Fig. 7.7) has closely approached the value it would assume after infinite time has elapsed. Our computations were stopped at T = 10 when the drag coefficient had reached a value of 1.71, which is close to the experimental value of 1.68 (Tritton, 1959). A time step $\delta t = 0.02$ was used in these simulations and at T = 10 approximately 30000 vortices existed in the wake. A cutoff vorticity value of 10^{-6} has been used. The drag exhibits a significant drop at the onset of the impulsive flow but does not change significantly after the appearance of secondary vorticity. The unsteady computations exhibit the generation of the secondary vorticity at T $\approx$ 1.5 (Fig. 7.9). The secondary vortices initially grow and subsequently remain unchanged as they are confined by the primary vortices. Each primary vortex (Fig. 7.10) is passively convected by the free stream velocity and as viscosity acts its vorticity is spread-out and does not create further secondary separation effects. Although its core is diffused each primary vortex is never detached from the body as a thick shear layer continuously feeds the primary vortex uninterrupted by the secondary separation effects.

In the second stage of this simulation the cylinder is **impulsively stopped**. The initial condition for this flow is the vorticity field as established at T = 10 by the previous impulsive start. As one may observe in Fig. 7.8 this impulsive stop induces a large inertial force on the body in opposite direction to the drag. A square root type singularity is exhibited by this flow as well. The body force eventually decreases to zero remaining always negative after the impulsive start. In Fig. 7.11 one may observe that immediately after the impulsive stop a strong vortex layer of the same sign as the secondary vorticity is developed on the surface of the body. This new layer then separates the primary vortex from the body and as it grows the magnitude of the drag is reduced. The pair of primary vortices forms a dipole that starts to move upstream. However its strength is not sufficient for it to travel far and the pair is

eventually stopped as it diffuses and is stabilized by the presence of the vortex layer of opposite vorticity. At T = 15 the vortex field has reached a configuration that essentially does not evolve further by convective effects and is subsequently eroded by diffusion.

- **Re = 550**

As the Re increases the flow structure becomes more complicated and new features of the vorticity field are exhibited. In Fig. 7.16 the streamline history of the flow field is presented. Note that only half of the domain is plotted, but no symmetry was imposed in our computations. One may observe the appearance of a small secondary region as well as the development of the dividing streamline of the flow. The development of the flow is alternatively described by the equivorticity plots of Fig. 7.15. Although at streamline patterns show the appearance of a secondary vortex at $T \approx 3$ secondary vorticity is already visible in the vorticity plots at T = 1. After its initial appearance the secondary vorticity remains confined and its evolution is mainly affected by the dynamics of the primary vortex. As the secondary vortex grows it penetrates the primary vortex but it is never able to reach the outer irrotational flow field. The primary vortex moves away from the body and its strength is reduced by diffusion stopping the growth of the secondary vortex. Initially the layer that feeds the primary vortex changes angle of orientation in respect with the body but it seems that a stable configuration (the so called α - phenomenon in *BC*) has been reached beyond T = 5.

In Fig. 7.13 we compare the instantaneous streamlines obtained from these computations with the streaklines obtained by *CB*. The results are in good agreement. However such good agreement in the streamline pattern has been observed in the past by other computational schemes as well (e.g., those of *TL*, *SS* etc.). In Fig. 7.14 the computational results of *CC*, *TL* and *SS* are presented. They are in good agreement with the experimental results as well (note however a smaller secondary region computed by *CC* and some discrepancies with the *SS* computations). However if one examines the drag coefficient obtained by different schemes there are several discrepancies. In Fig. 7.12 the drag coefficient obtained by the present numerical method is compared with the results of *CC* and *TL* as well as with the theoretical predictions of *BY* and *CD*. The present method accurately captures the transient behaviour and is in good agreement with the results of *CC*. However a large discrepancy is observed with the results of *TL* especially at early times. This may be attributed to the initial

flow condition *TL* uses for the impulsively started flow. Instead of the potential flow the viscous steady state solution at Re = 5 has been used. So initially in this approach vorticity is present throughout the domain contrary to a truly impulsive start. The moral of the story is that diagnostics such as streamlines allow for deviations from the correct solution, as they are a smooth function, and should not be blindly trusted as indications of the validity of an approach. On the other hand vorticity fields and drag coefficients are diagnostics that are two derivatives less smooth than streamlines and their accurate capturing substantiates the robustness of our code. Lecointe and Piquet (1984) found that the vorticity is not accurately tracked by their high order finite difference schemes even when a relatively accurate description of the streamlines is obtained.

The interplay of primary and secondary vorticity is manifested in the drag curve. After the initial drop in the drag the appearance and growth of the secondary vortex increases the drag coefficient as it pushes outwards the primary vortex layer. This increase reaches a maximum at T = 3 beyond which the strength and the size of the secondary vortex are reduced whereas the primary vortex is further convected by the free stream and the drag decays to its steady state value (assuming symmetry persists).

At the a second stage of this computation the flow is **impulsively stopped**. However, in contrast with the Re = 40 case, the primary and secondary vortices are stronger resulting in a more complicated flow configuration. In Fig. 7.18 the time history of the vorticity field after the impulsive stop is presented. Due to inertial effects a strong layer of secondary vorticity is formed on the vicinity of the boundary that carries along the secondary vortex already present in the rear of the cylinder. Simultaneously the dipole of the primary vortices is moving upstream due to the absence of the balancing velocity component of the free stream. Selected vorticity fields from this flow are presented in Fig. 7.19 and Fig. 7.20 In Fig. 7.17 the drag coefficient initially becomes negative but increases rapidly enough to reach a maximum at T = 6.0. After this point most of the vorticity field has been carried upstream of the cylinder and tertiary vortices are formed on its surface. The secondary vortex is rearranged and eventually at T = 9 reaches the outer flow and at this point the drag coefficient reaches a minimum. Subsequently new dipoles are formed around the cylinder and the vorticity field decays by the act of diffusion, as essentially no new vorticity is generated on the surface of the cylinder, and the drag decays to zero.

One may observe that although such dramatic changes occur in the vorticity field the flow remains symmetric and this symmetry is not sensitive to roundoff errors present in the computations.

- **Re = 1000**

The time history of the streamlines for the case Re = 1000 is shown in Fig. 7.23. Phenomena similar to the ones observed in the previous case are present here as well. The primary vortex is formed at the rear of the cylinder and the secondary vortex, although stronger in this case, remains confined by the primary vortex. As it may be observed in Fig. 7.22, showing equivorticity lines, the secondary vortex attempts to reach the outer flow field but is strained by the primary one before cutting the feeding link of the primary with body. At the end of the simulation a tertiary vortex is discerned on the surface of the body.

The parameter to observe here is the drag coefficient shown in Fig. 7.21. For this particular Re several computational results are available to be compared with the results of the present scheme. The present vortex method captures fairly well the initial square root singularity of the drag coefficient and is in agreement with the theoretical results of *BY*. The agreement with the method presented in *CC* is still satisfactory although some discrepancies appear around the minimum and maximum of this curve (when vorticity is most active). This discrepancy may be attributed to the additional viscous dissipation present in the scheme of *CC* that reduces the actual strength of the vortices.

Large discrepancies appear however with the results of the present method and those presented by *TL* and *SS*. The effects of the initial condition selected by *TL* become more pronounced at this higher Re. The flow of *TL* seems to never recover to that corresponding to an initially impulsively started cylinder. On the other hand the scheme discussed by *SS* presents an erratic behaviour as well. The initial transient behaviour is not captured (giving even zero and negative drag) demonstrating the problems of the algorithm to simulate vorticity generation due to solid walls and the low convergence rate of the random walk approach to model viscous effects. At later times the drag coefficient oscillates signifying dynamics not present in all other computations.

- **Re = 9500**

The case Re = 9500 is the highest Re for which simulations were carried out in this thesis and by far the more interesting and challenging one. The time step used is $\delta t = 0.0125$, the cutoff vorticity limit is kept to 10^{-5} and at the end of the simulation at T = 4.0 *more than half a million vortices* are used to capture all the scales present in the flow. The flow eventually becomes asymmetric due to the accumulation of roundoff errors.

In Fig. 7.28 the streamlines of the flow are presented for a series of times outlining the complex structures present in the flow field. The picture becomes more vivid however if one examines the vorticity field of the flow as presented in Fig. 7.29. Initially a very thin layer of vorticity resides around the cylinder as it is introduced by the initial potential flow. This thin layer continues to grow while maintaining a relatively invariable form. In the same time the square root singularity is observed in the drag coefficient. However after T = 1.5 a bulge of vorticity forms at $\theta = 45^o$ that modifies the dynamics of the wake. This bulge constitutes now the center of the primary vortex under which a secondary vortex is formed. However this secondary vortex is formed at a much larger angle from the rear stagnation point than in lower Re. As it is also stronger it penetrates the feeding vortex sheet of the primary vortex and reaches the outer flow. The primary vortex then rolls up and detaches eventually from the body carrying along a part of the secondary vortex as shown in Fig. 7.29d. As the secondary vortex is severed its strength reduces and a new positive bulge of vorticity, created by the shear layer, comes to suppress it. This new vortex is formed at higher angle yet extracting some positive vorticity from the body as well so that the secondary vortices have a wave like shape. As the new vortex moves downstream, it reaches the primary vortex and merges with it reestablishing the link of the primary vortex with the body. The resulting vortex evolves and as its strength has not been diminished significantly by diffusion, spins around itself therefore reducing again the width of the vorticity link with the body. During this process tertiary vortices appear on the surface of the body but they are relatively weak and do not seem to affect significantly the evolution of the secondary vortex that confines them.

The interplay of the primary and secondary vortices do not remain unnoticed by the drag coefficient. The drag after its standard initial drop increases as the primary vortex continues to grow. This increase is stopped however at T $\approx$ 2.0 when the secondary vortex reaches the outer flow and the primary vortex is detached (Fig. 7.30). It remains relatively constant for a short time and continues to increase

when the feeding of the primary vortex is reestablished. It reaches a maximum and then drops as new vorticity is created on the surface of the body. The discrepancy of the present results and those presented by *CC* is more pronounced. More specifically the scheme of *CC* does not predict the drag plateau that appears at T $\approx$ 2.0. It seems that the additional numerical diffusion present in that scheme suppresses the communication of the secondary vortex with the outer flow and allows the feeding of the primary vortex and the drag increase. The double bump in the drag coefficient seems to be present at simulations at lower Re numbers (Re = 3000) as reported by Pépin (1990) and may be associated there as well with a similar interplay of primary and secondary vorticity.

7.2.4.2 The translating and rotating cylinder

In this section the flow resulting from a cylinder set into translation and rotation simultaneously is discussed. The initial stages of the flow are examined to validate our numerical approach and determine its ability to simulate general unsteady flows. When α is nonzero a lift force is exerted on the cylinder and non zero circulation enters the fluid. The cylinder movement plays an important role in the formation of the wake. The mechanics of the boundary layers are different on the top and bottom side of the cylinder as on top the newly created vorticity is initially convected counter to the free stream and on the bottom is pushed farther by the cylinder rotation.

The results presented in this section are concerned with the wake formation at very early times for Re = 200 and $\alpha = 0.5$. They are compared with available computational and experimental results to validate our method. In Fig. 7.33 equivorticity contours of the flow are presented. The wiggles that appear on these curves may be attributed to the distortion of the Lagrangian grid as a relatively large time step of $\delta t = 0.025$ was used in these simulations, as well as with the sporadic remeshing that has been performed. The flow at this early stage appears to be nearly symmetric, however one may discern the rotation of the secondary vortices on the rear of the cylinder and larger vorticity gradients at the top part of the flow. In Fig. 7.34 the computed streamlines are presented. The initial symmetric pattern has been broken and fluid from the lower part of the domain is pushed upwards due to the cylinder rotation. Subsequently a rotational and a hyperbolic point are manifested in the streamline pattern at T = 2 and a second vortex appears forming at T = 3.0.

The results of the present computations are compared with computational results in Fig. 7.35 *BD* and with flow visualization in Fig. 7.36 *CM*. It may be seen that a remarkable agreement exists between these results. In Fig. 7.31 the drag coefficient is presented. It is shown that the rotating cylinder exhibits a larger drag mainly due to the larger vorticity developed on the top of the cylinder. In Fig. 7.32 the lift coefficient is presented and is compared with the analytic results of *CD*. At the onset of the flow the computed results are not capturing very well the transient of the lift coefficient as a sign of poor discretization. A time step of $\delta t = 0.02$ (as that for a non-rotating case) has been used for this simulation.

A final quantitative comparison is made in for the u velocity component in the rear axis ($\theta = 0$) ant at the y^+ axis ($\theta = 90^o$) of the cylinder in Fig. 7.37 and Fig. 7.38 respectively. The results of the present computation are indistinguishable from those presented by *BD* and are in excellent agreement with the experimental results of *CM*.

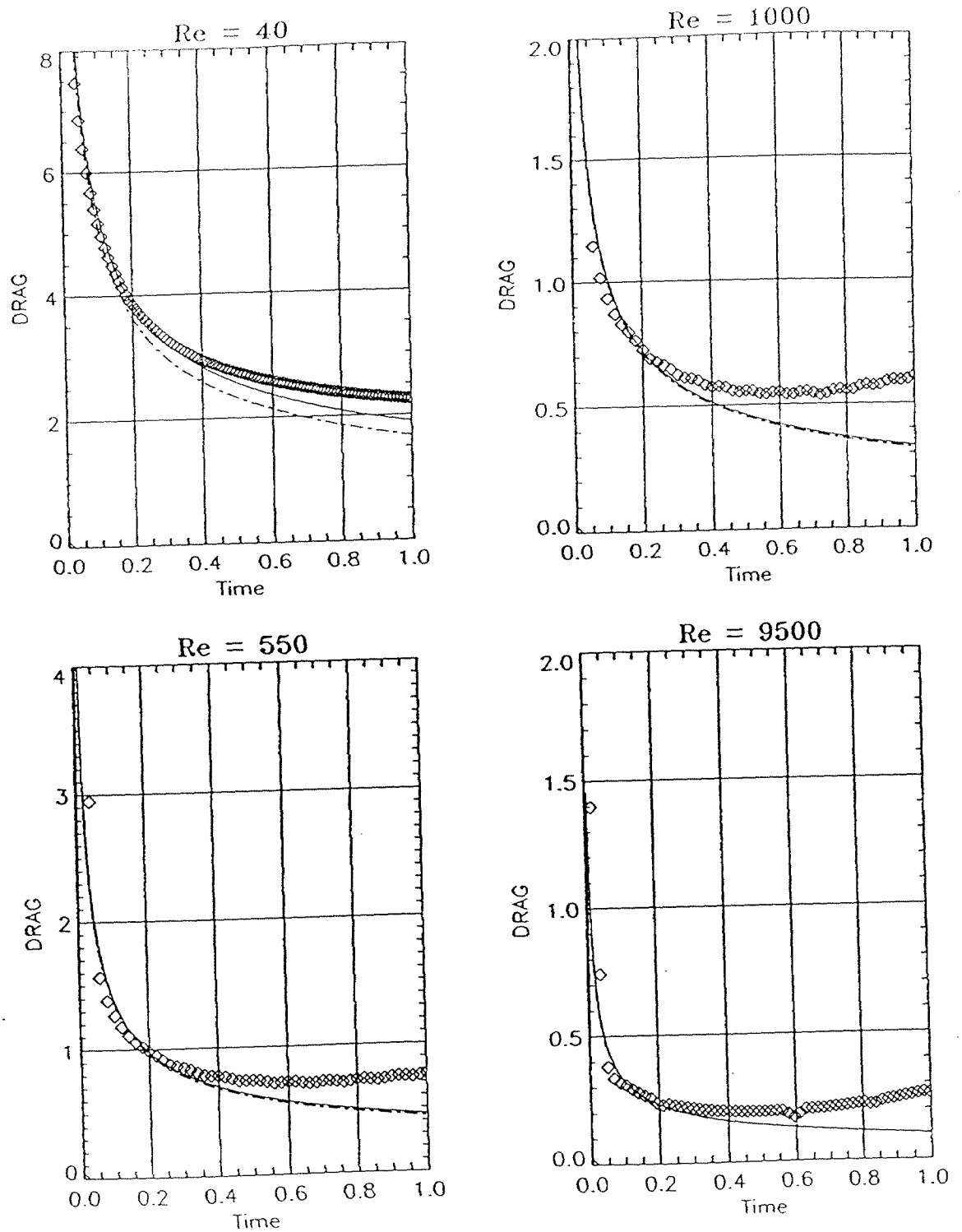


FIG. 7.5 Linear plot of the early time history of the drag coefficient for an impulsively started circular cylinder. Solid line (BY), dashed Line (CD), symbols (present computations)

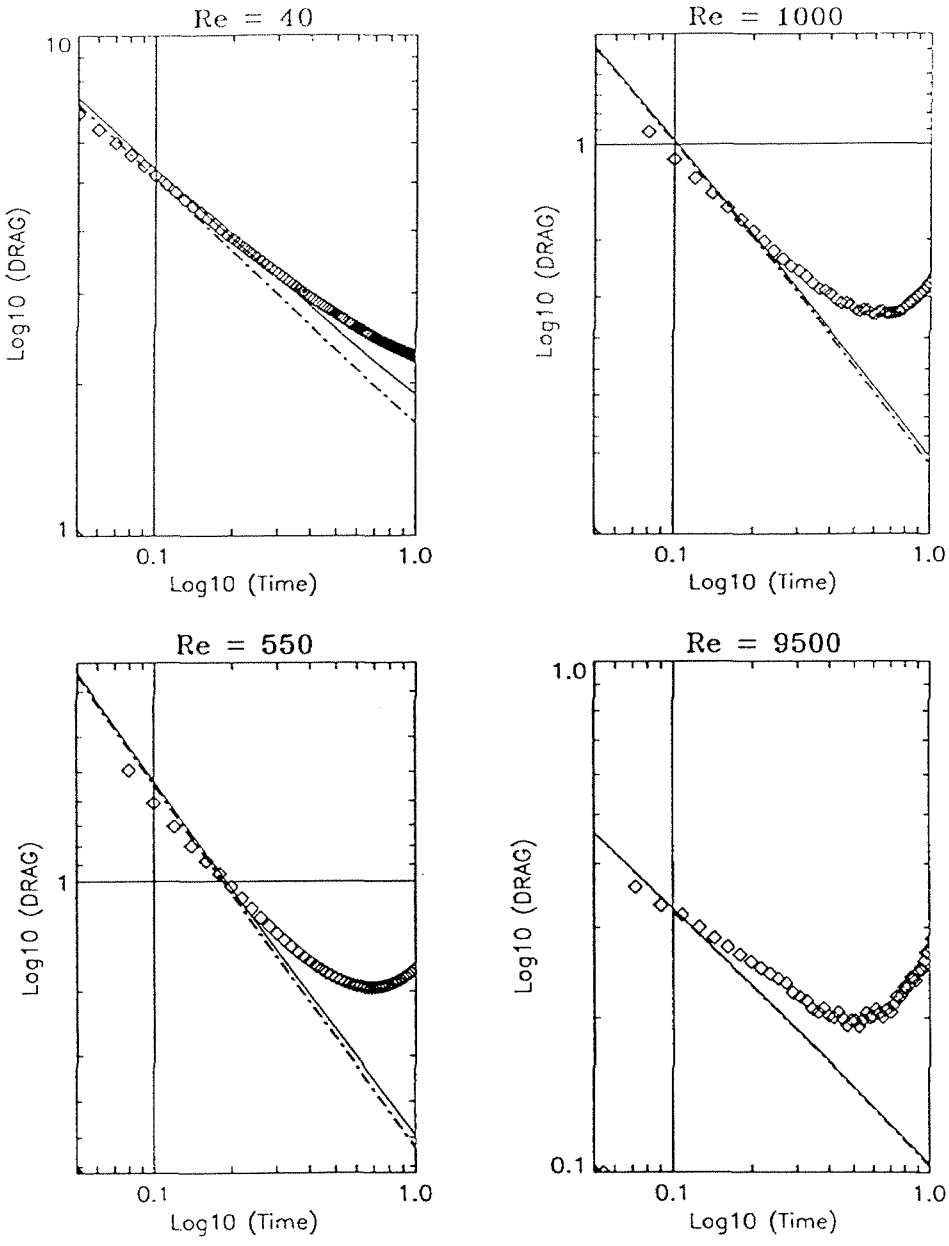


FIG. 7.6 Logarithmic description of the early time history of the drag coefficient for an impulsively started circular cylinder. Solid line (BY), dashed Line (CD), symbols (present computations)

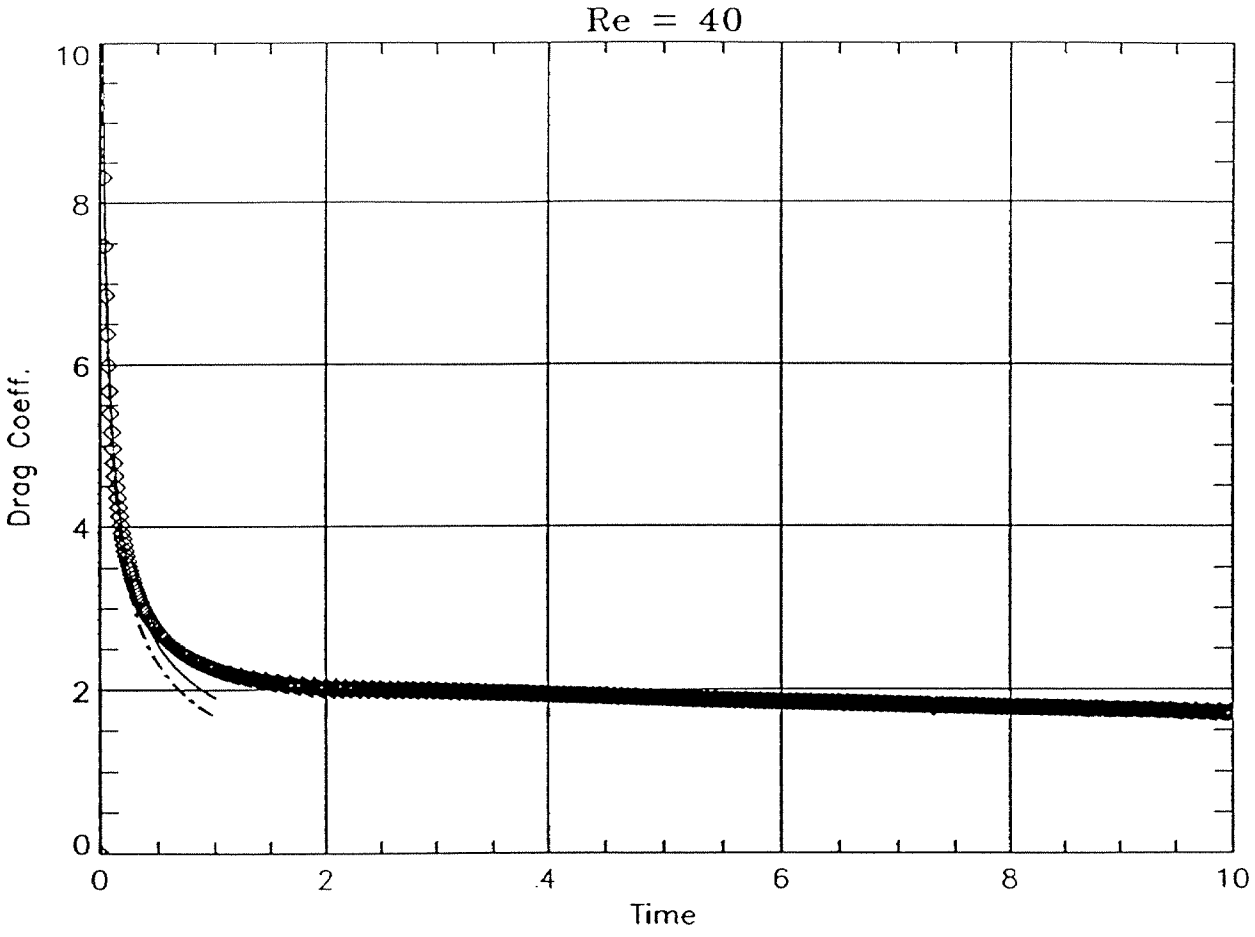


FIG. 7.7 Drag Coefficient for an impulsively started circular cylinder. Re = 40. Solid line (BY), dashed Line (CD), symbols (present)

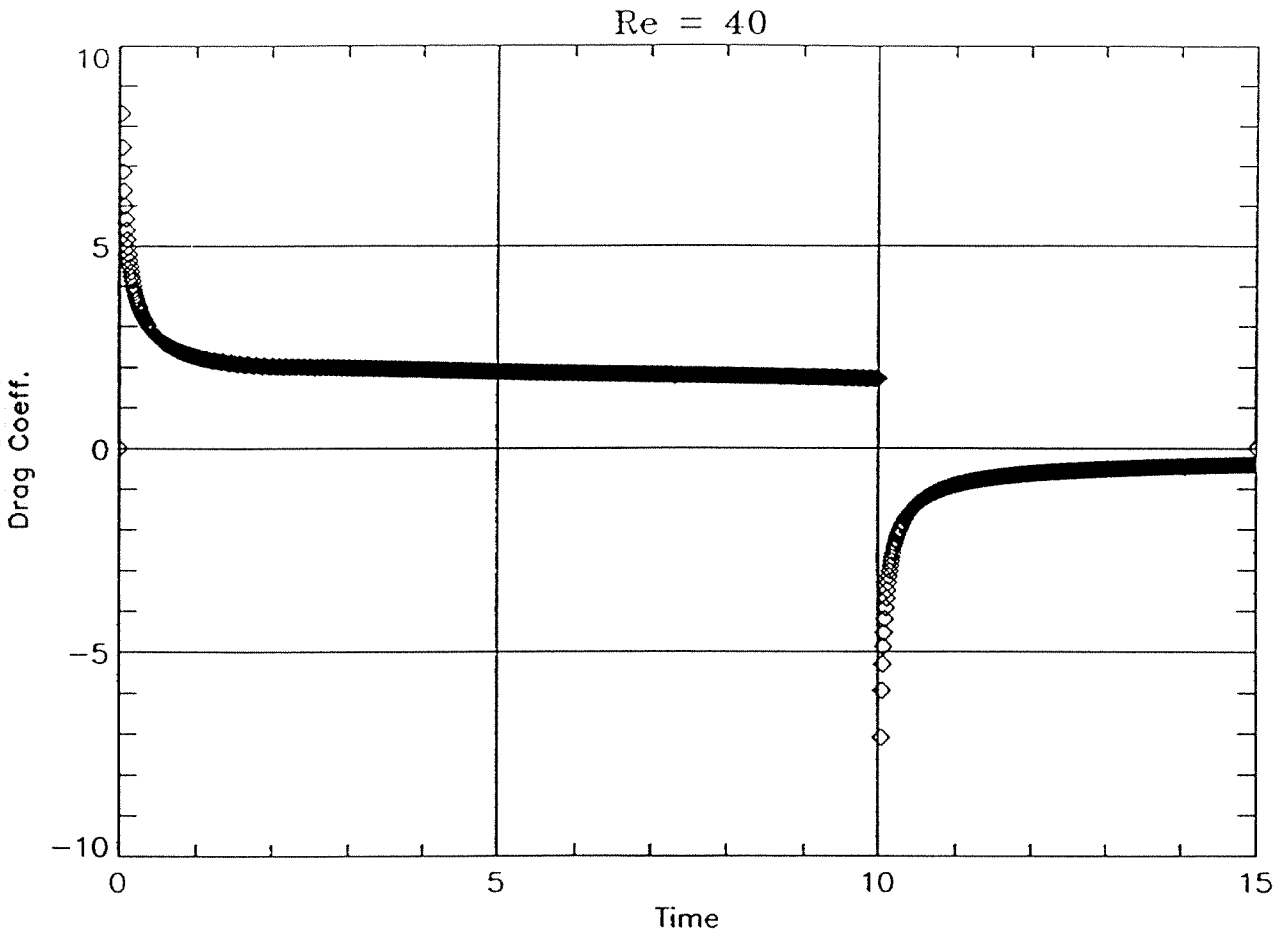


FIG. 7.8 Drag Coefficient for an impulsively started and impulsively stopped (at T = 10) circular cylinder for Re = 40.

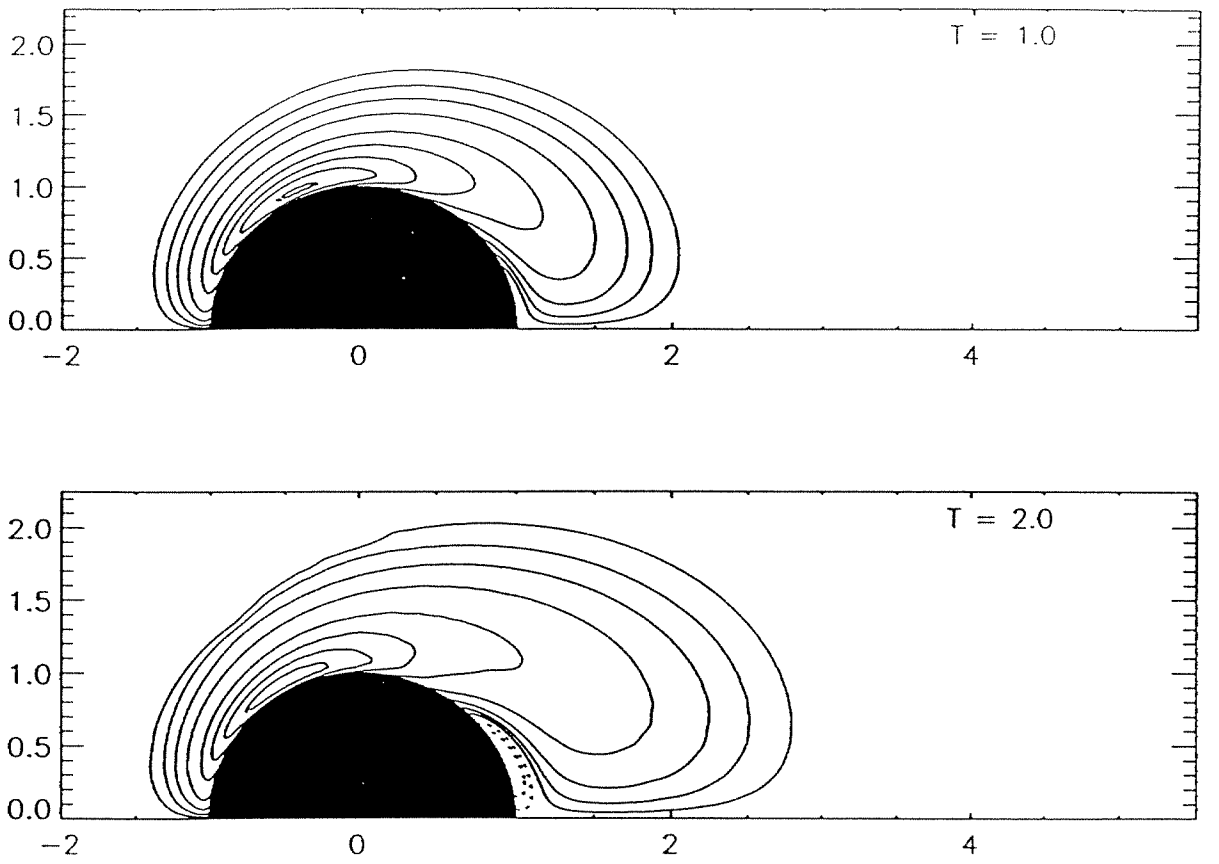


FIG. 7.9 Equi-vorticity contours for an impulsively started circular cylinder for Re = 40. Solid lines : negative (clockwise) vorticity. Dashed lines : positive vorticity.

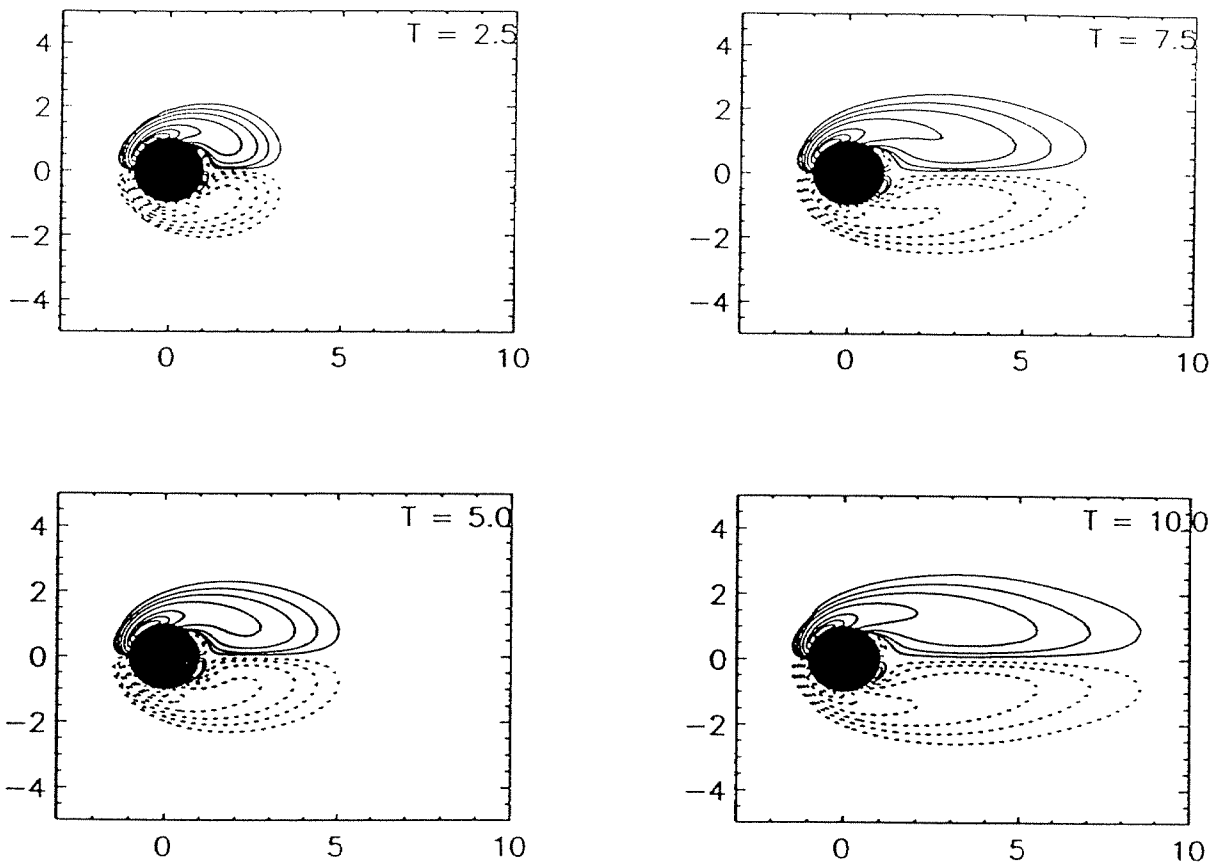


FIG. 7.10 Time history of e qui-vorticity contours for an impulsively started circular cylinder for Re = 40.

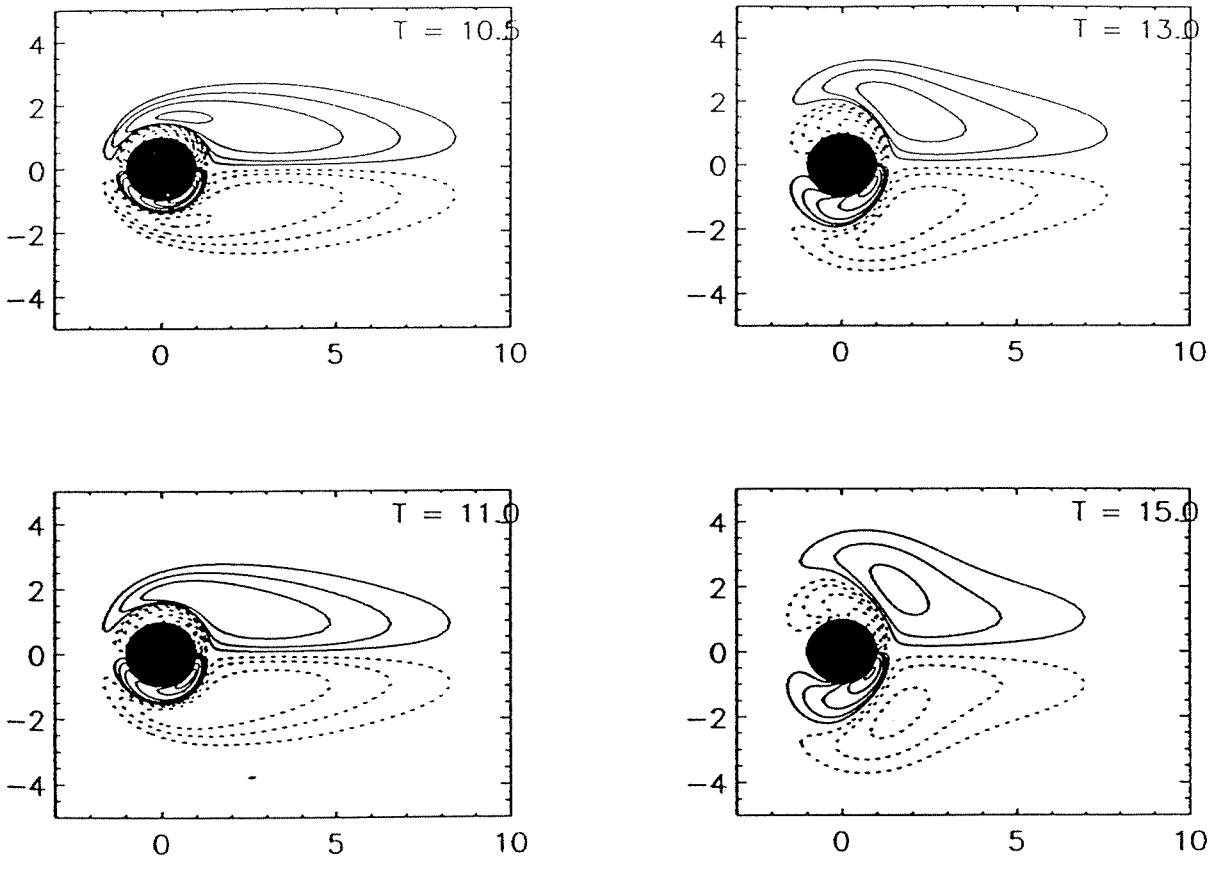


FIG. 7.11 Equi-vorticity contours after the impulsive stop (at T = 10) of an impulsively started circular cylinder for Re = 40.

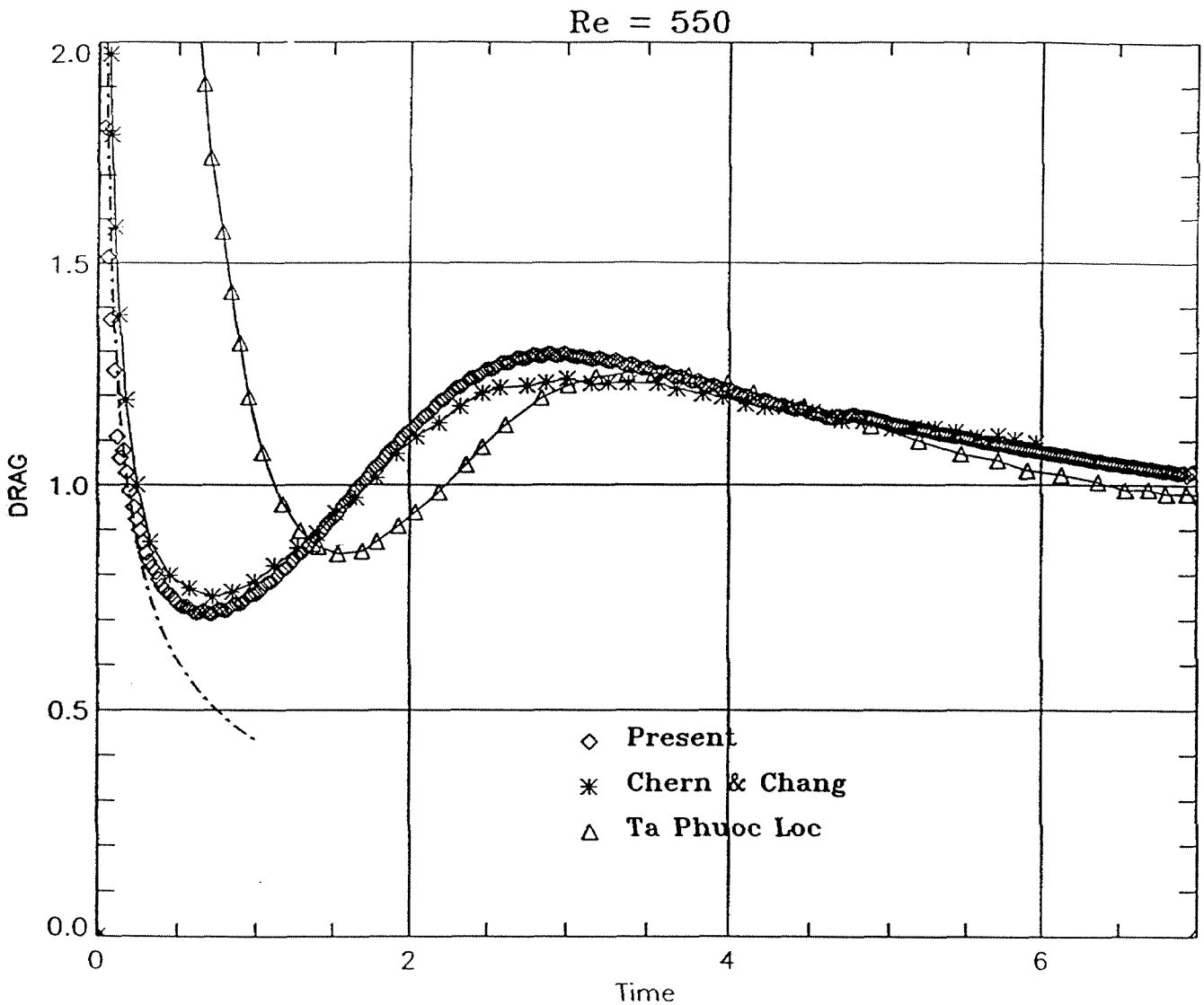


FIG. 7.12 Comparison of the drag coefficient of an impulsively started circular cylinder for Re = 550 as computed by several numerical schemes. Dashed line : analytical results (BY).

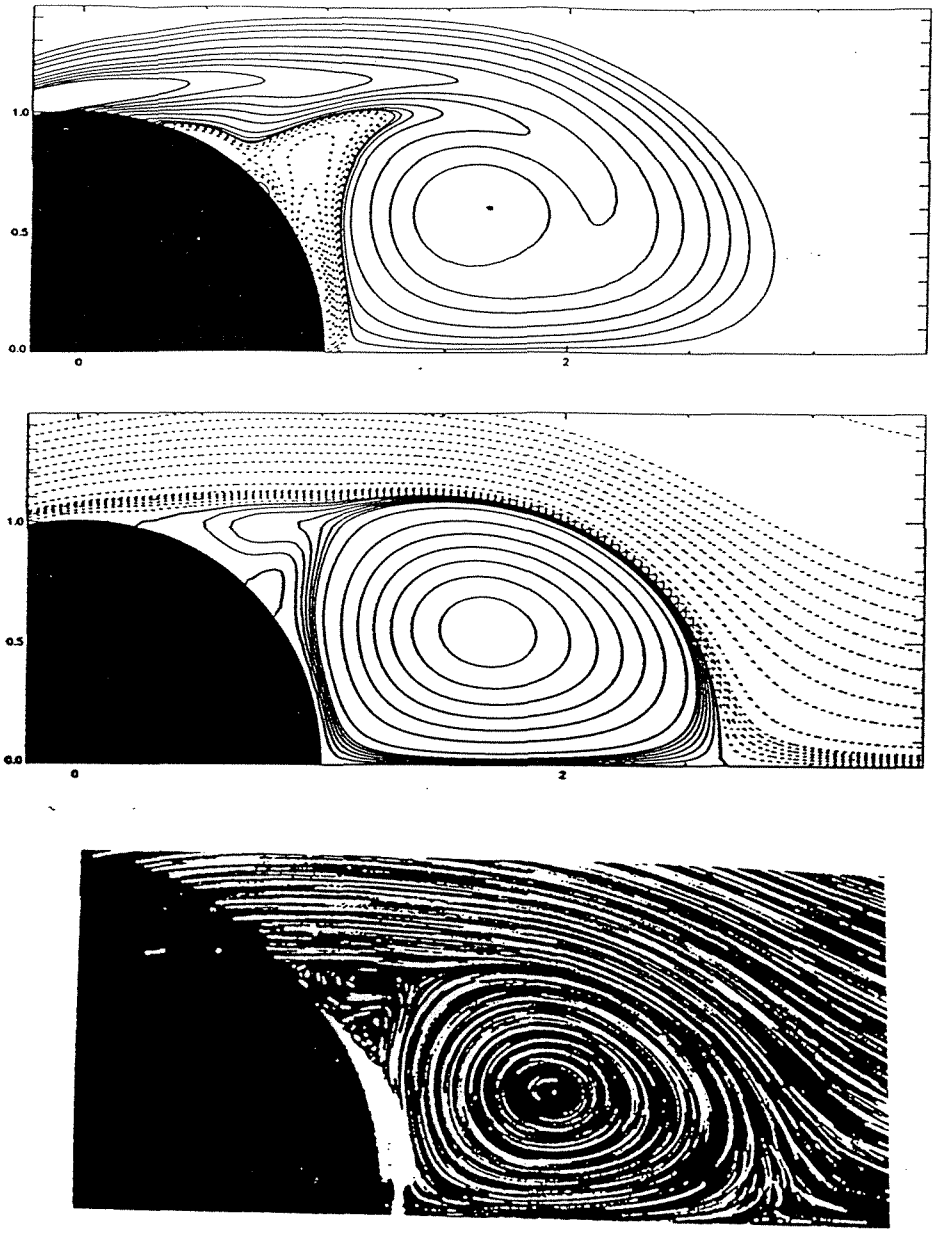


FIG. 7.13 Present computational results (equi-vorticity lines and instantaneous streamlines) and comparison with experimental results (streaklines) of BC for an impulsively started cylinder for Re = 550 at T = 5.

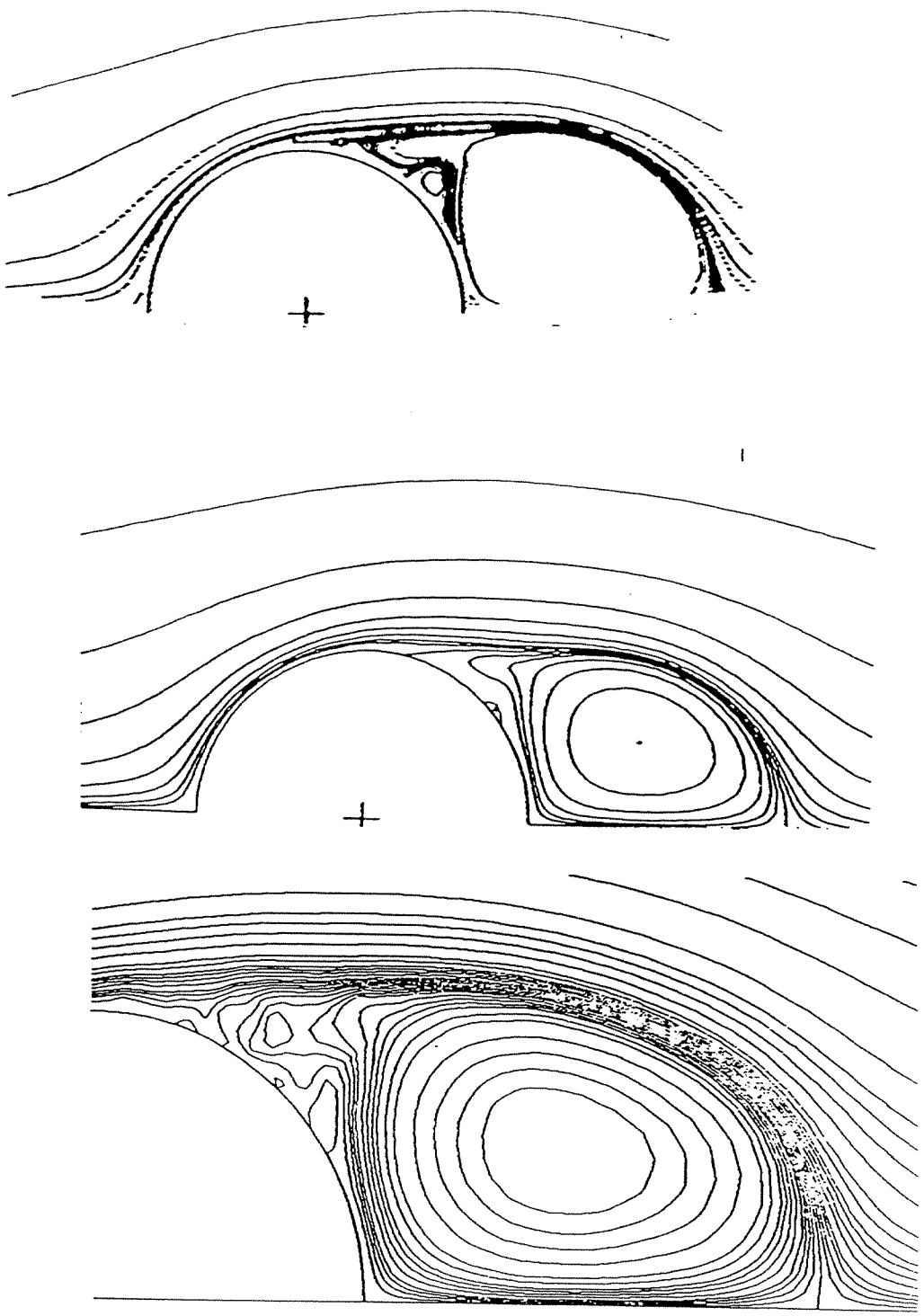


FIG. 7.14 Computational results of *TL* (top), *CC* (middle), and *SS* (bottom) for an impulsively started circular cylinder for Re = 550 at T = 5

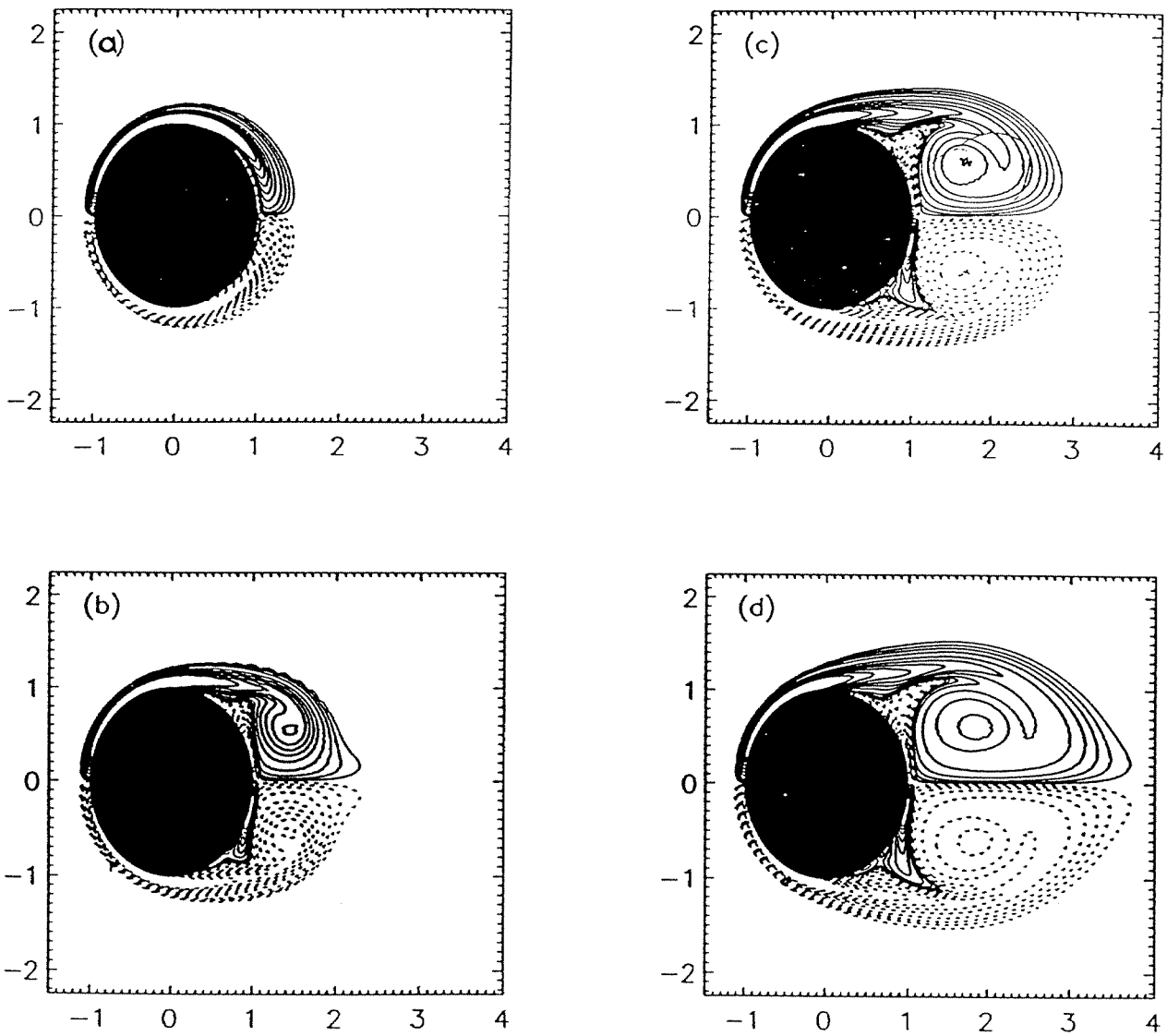


FIG. 7.15 Equi - vorticity contours for an impulsively started circular cylinder at Re = 550. (a) T = 1, (b) T = 3, (c) T = 5 (d) T = 7

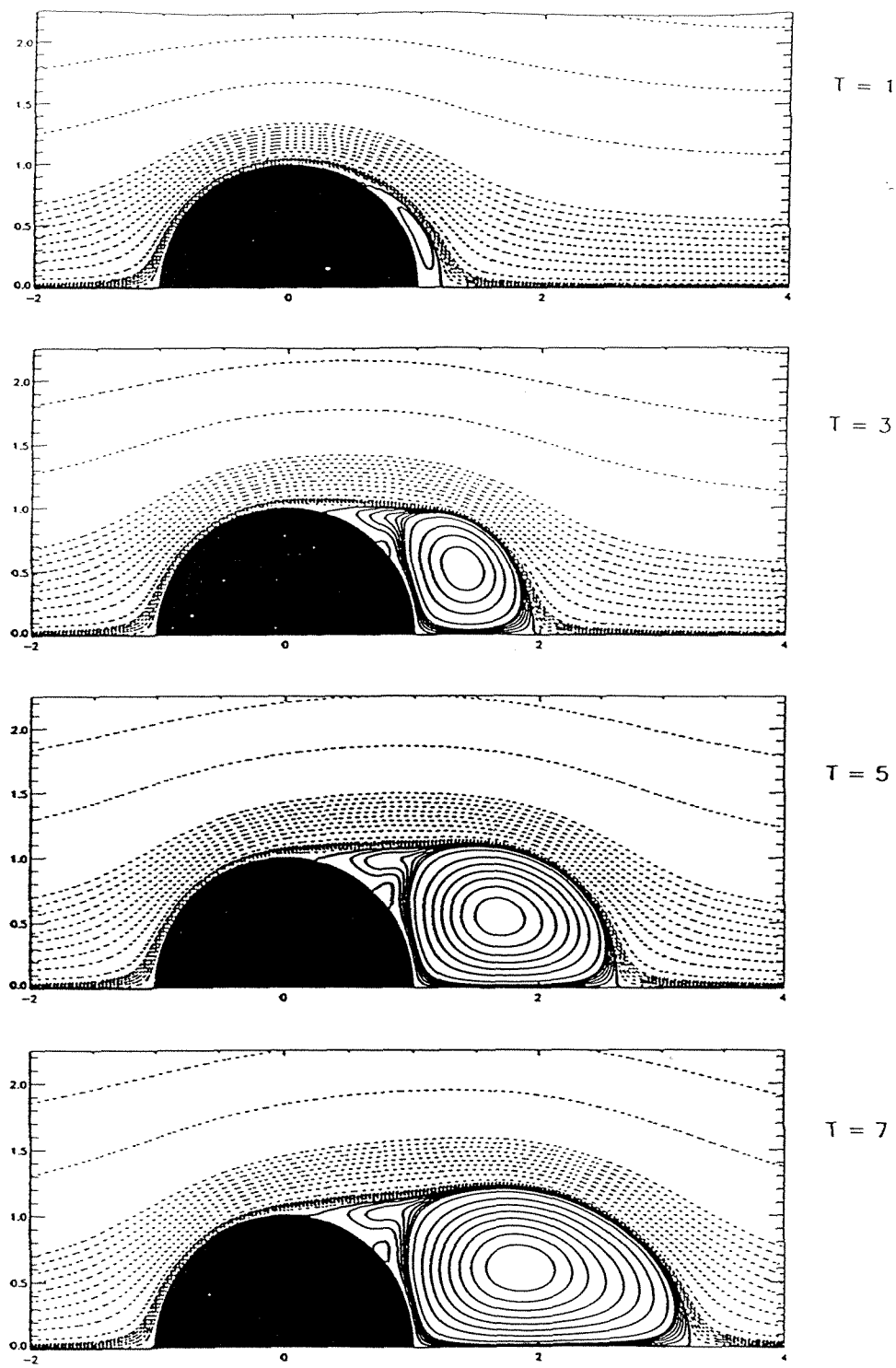


FIG. 7.16 Streamline time history for an impulsively started circular cylinder at Re = 550.

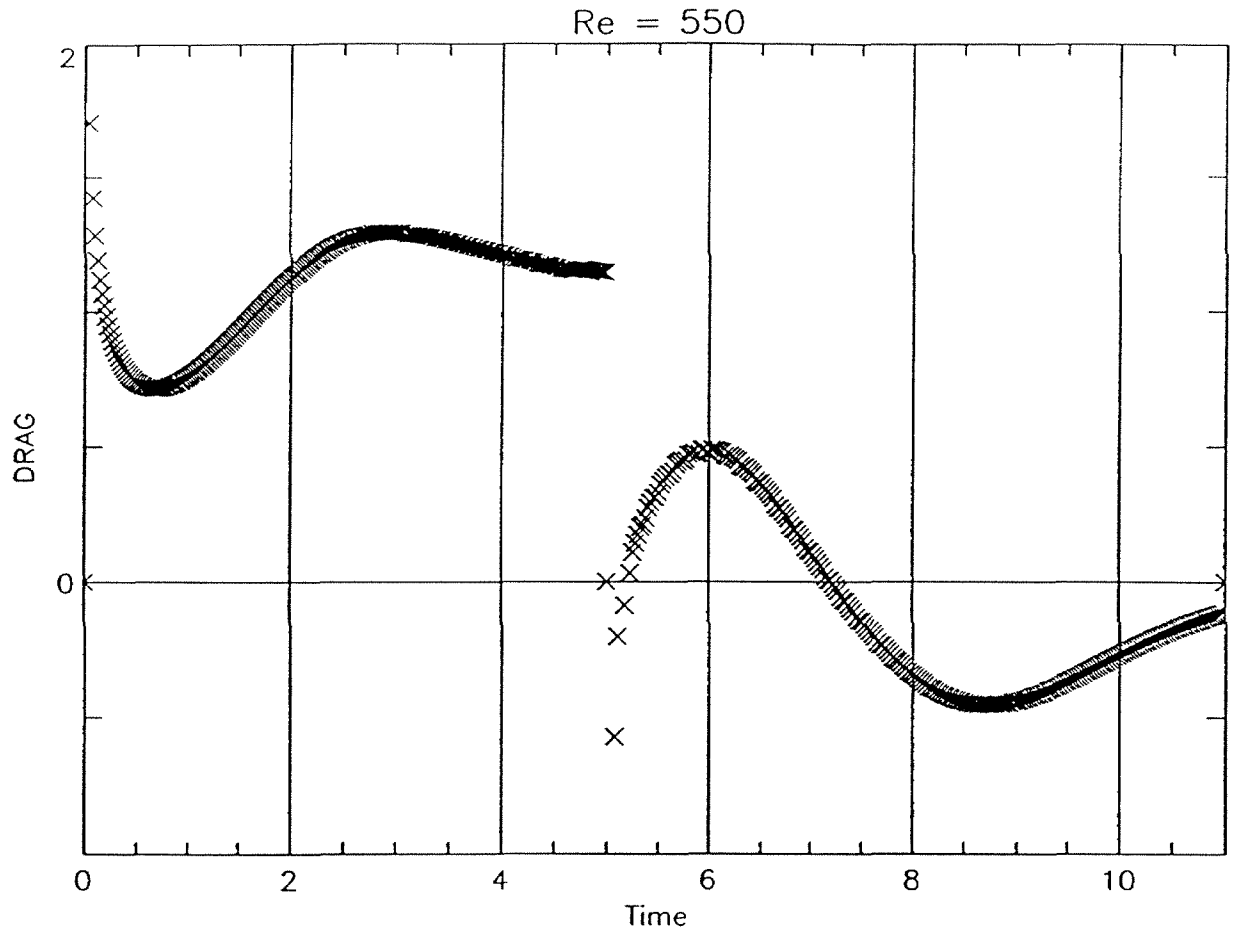


FIG. 7.17 Drag coefficient for an impulsively started and impulsively stopped (at T = 5) circular cylinder at Re = 550

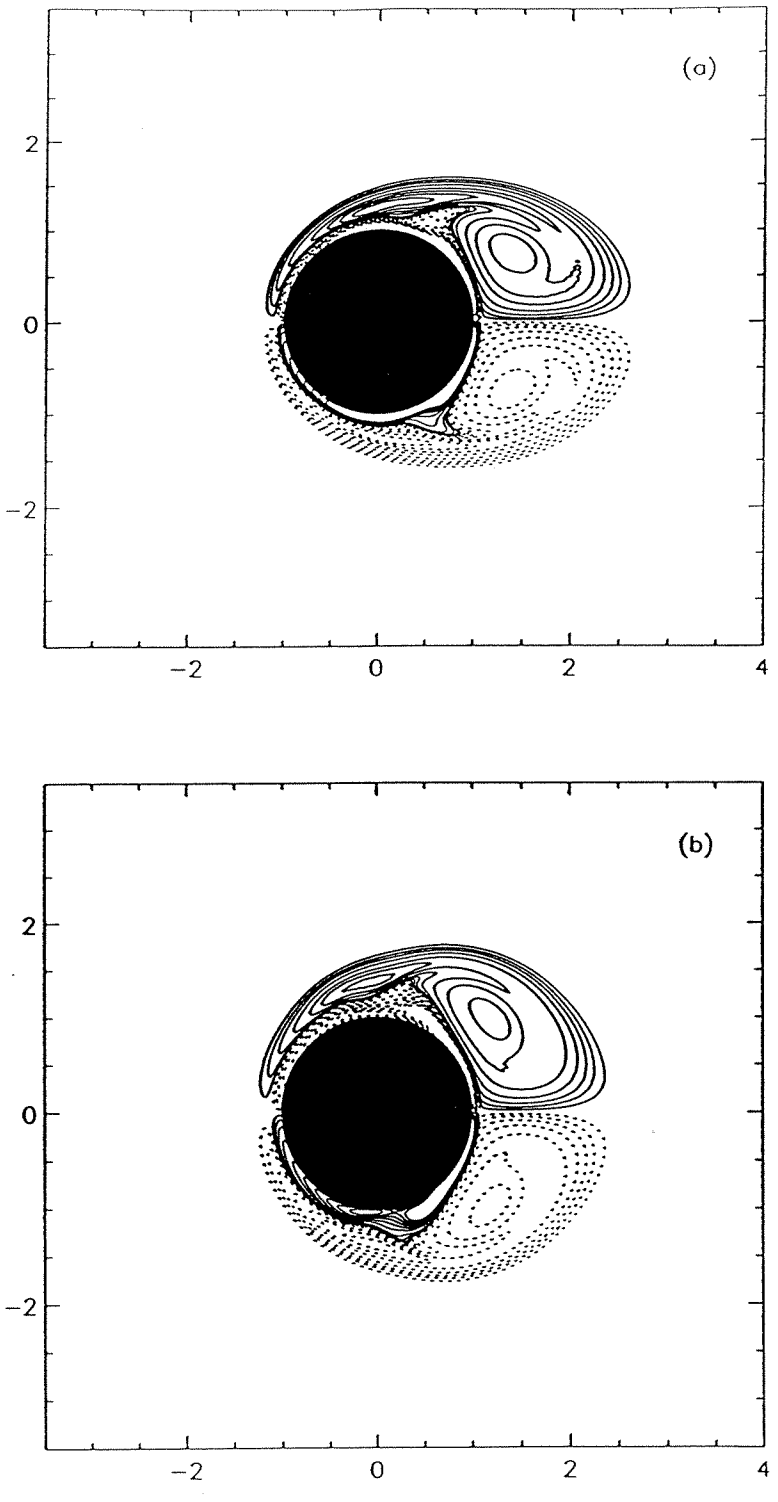


FIG. 7.18a Equivorticity plots after the impulsive stop (at T = 5) of an impulsively started circular cylinder (a) T = 5.5, (b) T = 6.0

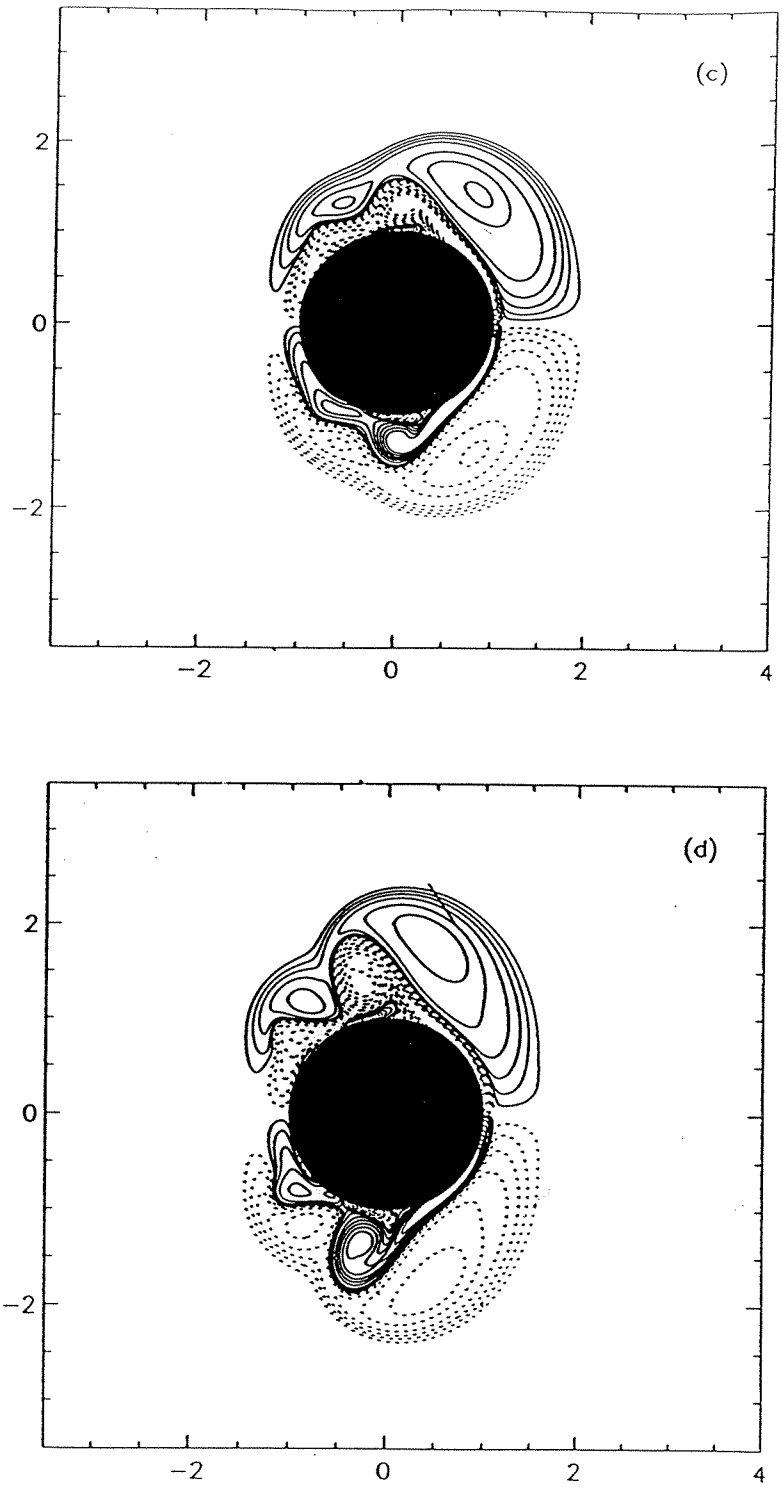


FIG. 7.18b Equivorticity plots after the impulsive stop (at T = 5) of an impulsively started circular cylinder (c) T = 7.0, (d) T = 8.0

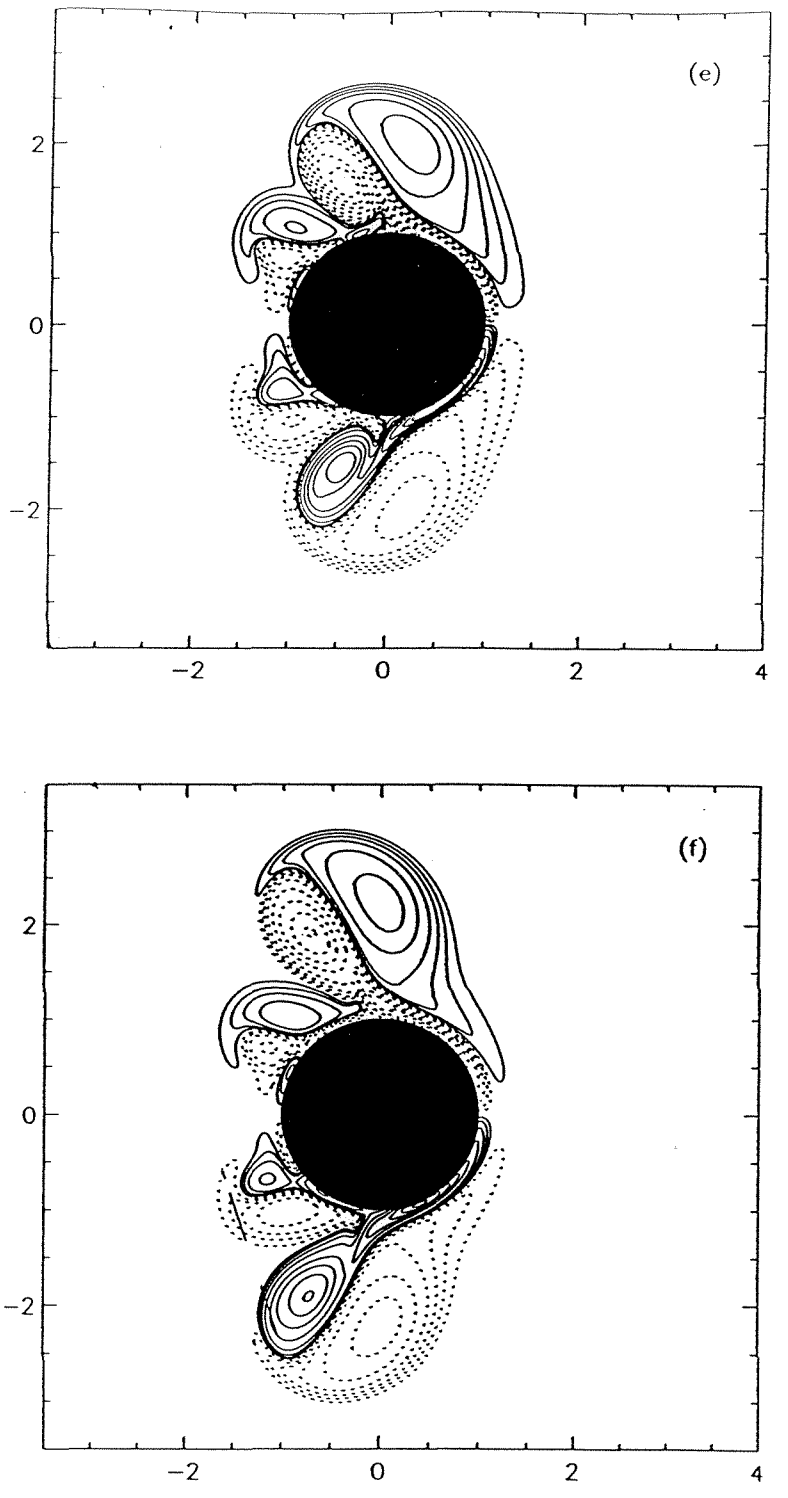


FIG. 7.18c Equivorticity plots after the impulsive stop (at T = 5) of an impulsively started circular cylinder (e) T = 9.0, (f) T = 10.0.

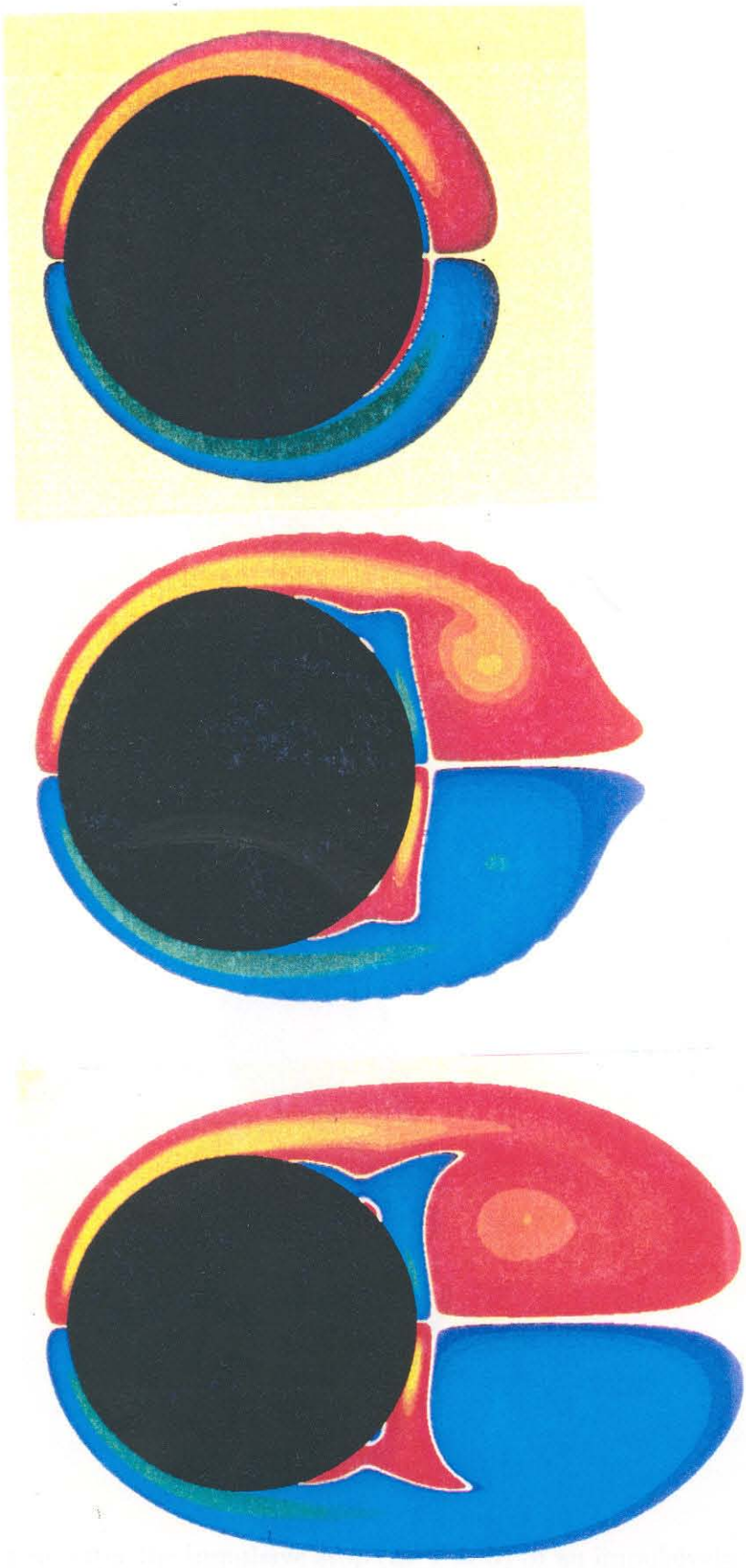


FIG. 7.19 Vorticity field of an impulsively started circular cylinder at Re = 550. T = 1.0 (top), T = 3.0 (middle), T = 5.0 (bottom)

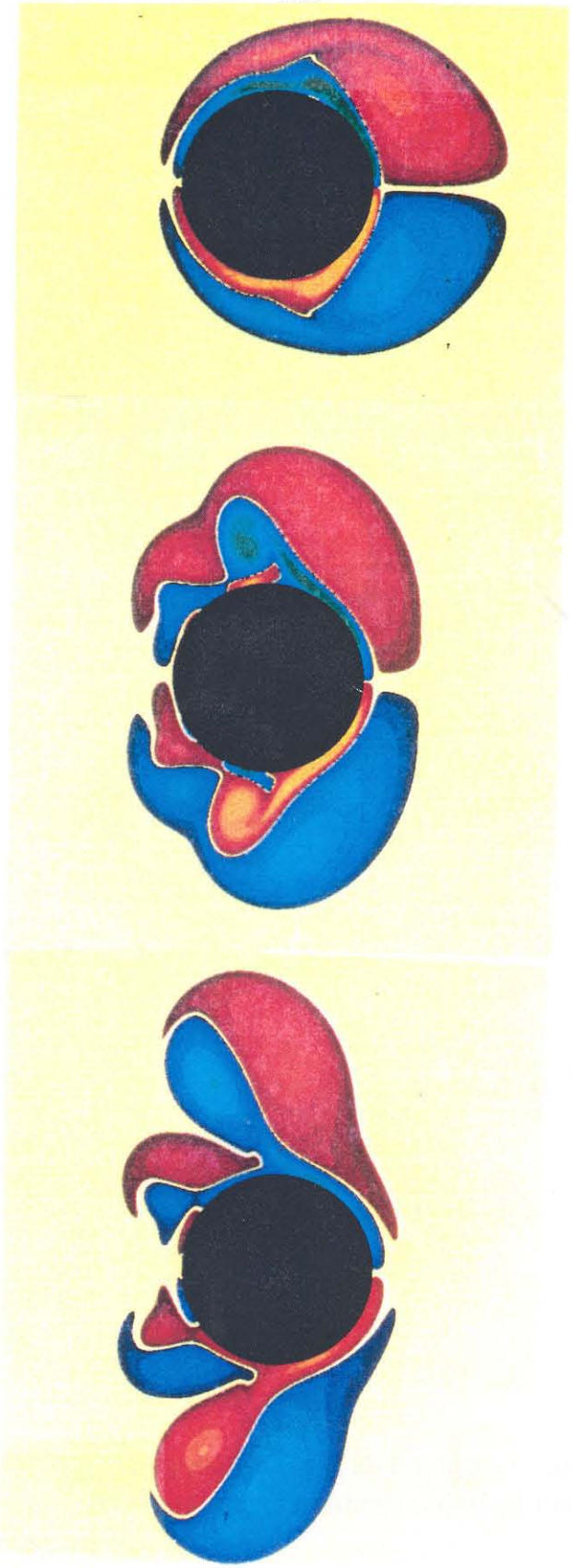


FIG. 7.20 Vorticity field after the impulsive stop (at T = 5) of an impulsively started circular cylinder at Re = 550. T = 6.0 (top), T = 8.0 (middle), T = 10.0 (bottom).

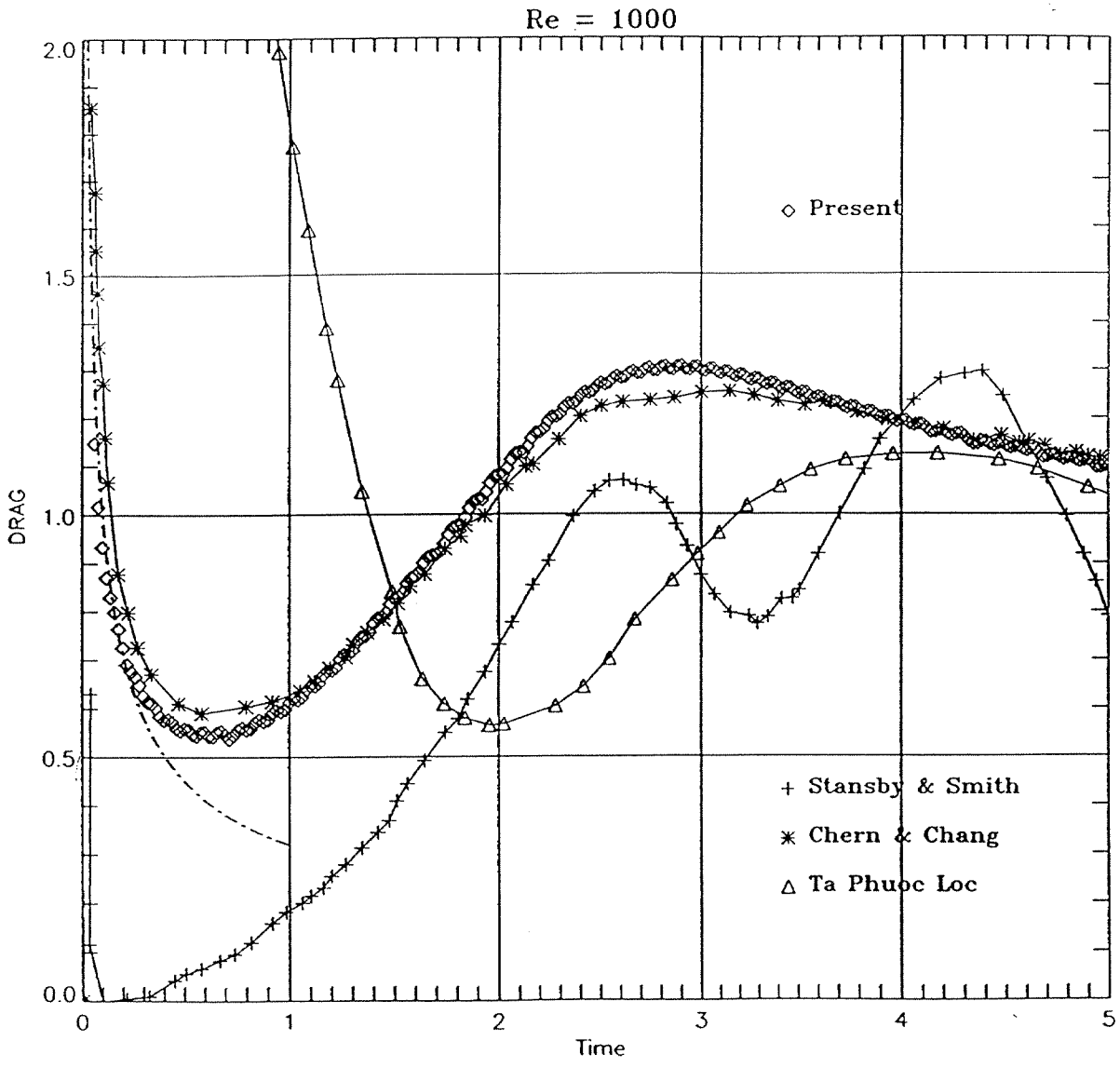


FIG. 7.21 Comparison of the drag coefficient of an impulsively started at Re = 1000 as computed by several numerical methods. (dashed line : analytical results of BY)

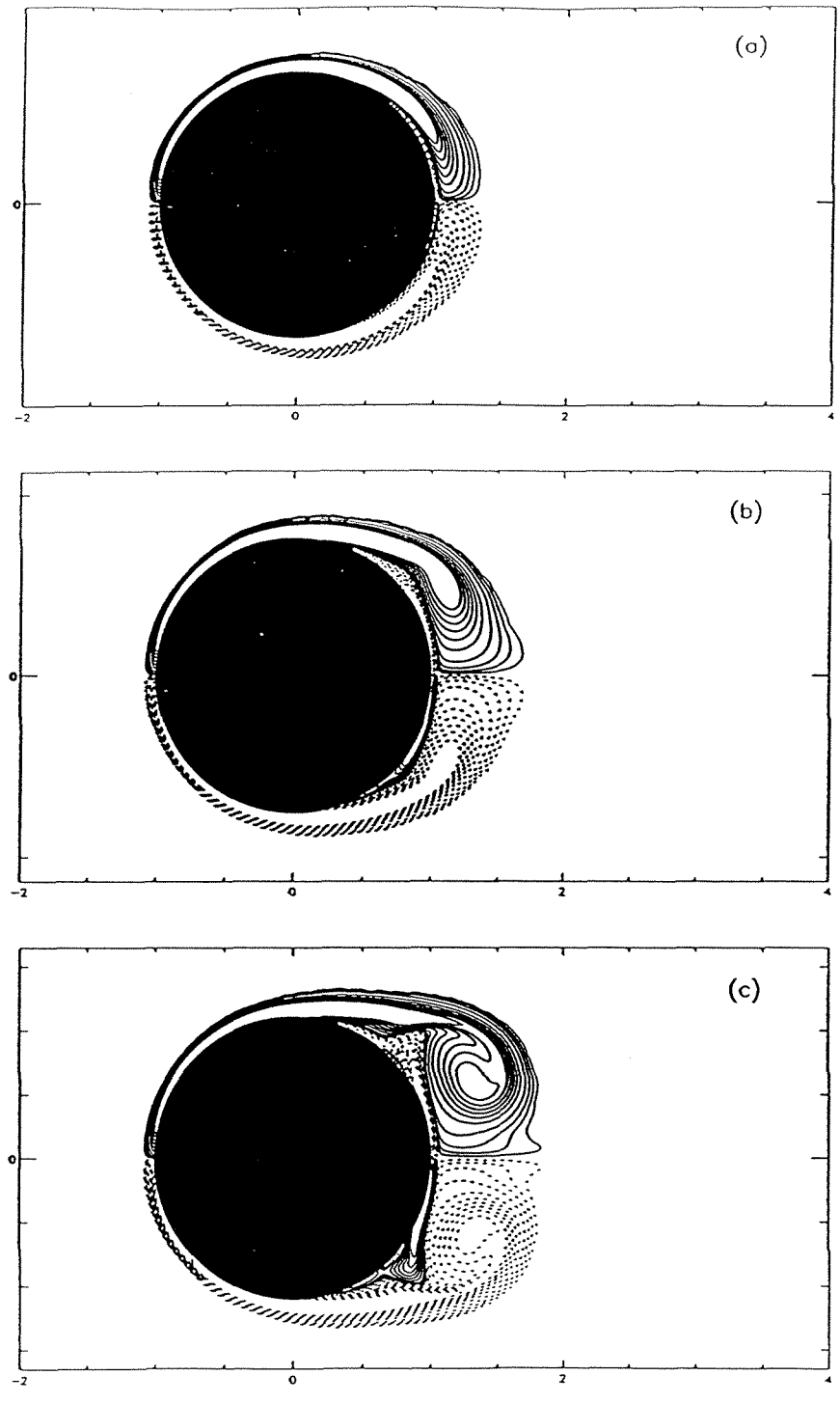


FIG. 7.22a Equivorticty lines of an impulsively started circular cylinder at Re = 1000. (a) T = 1.0, (b) T = 2.0, (c) T = 3.0

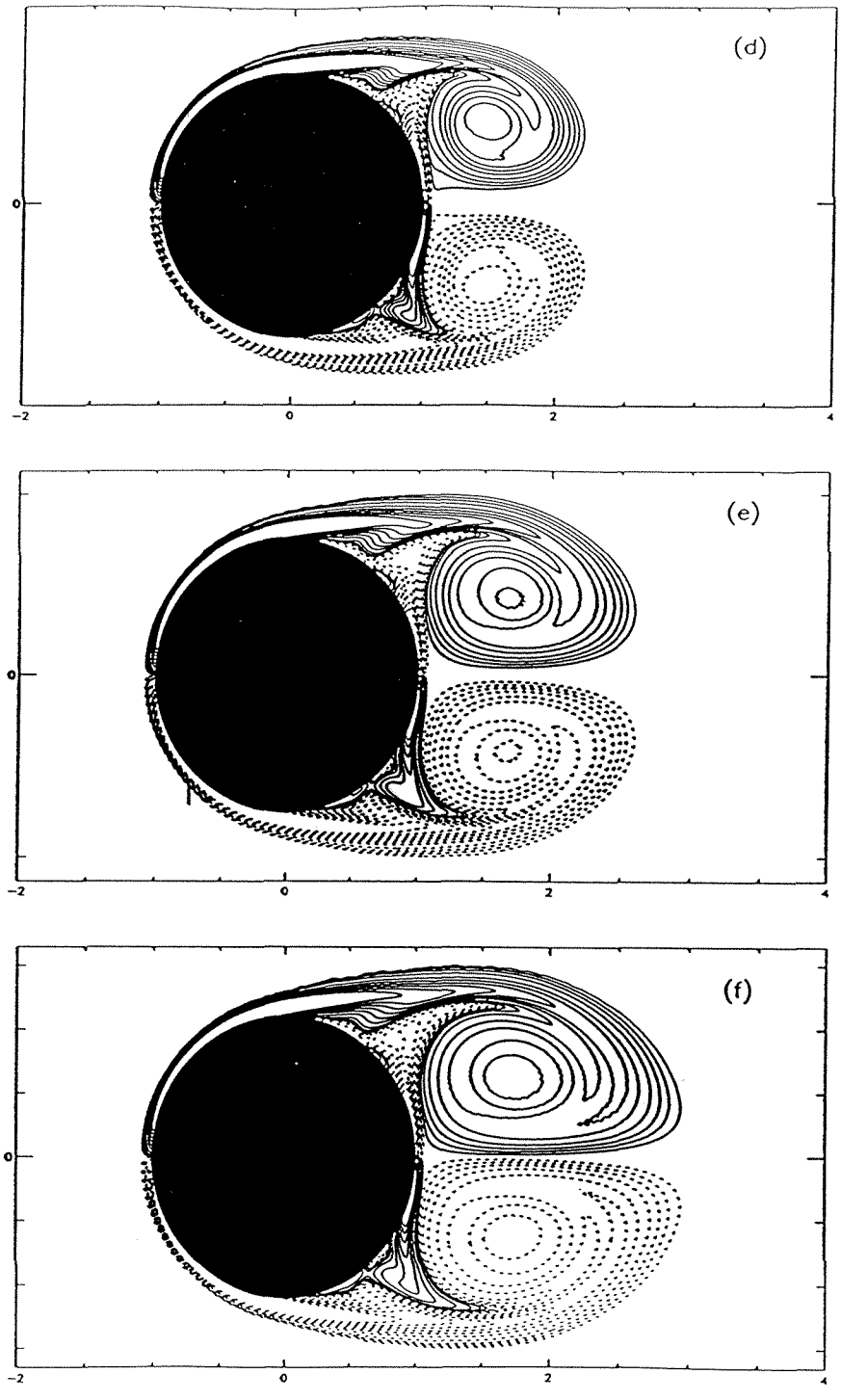


FIG. 7.22b Equivorticty lines of an impulsively started circular cylinder at Re = 1000. (d) T = 4.0, (e) T = 5.0, (f) T = 6.0

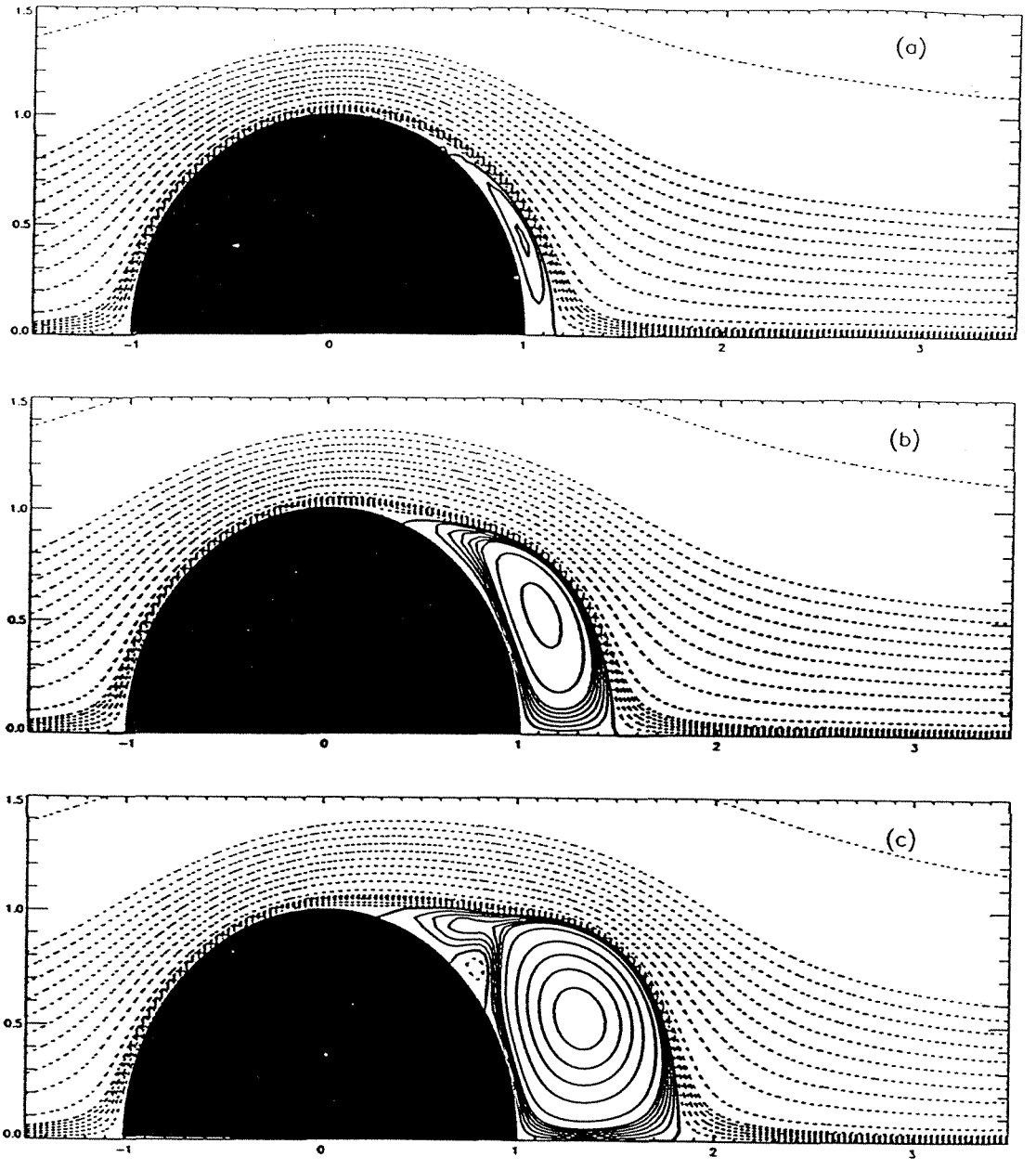


FIG. 7.23a Instantaneous streamlines of an impulsively started circular cylinder at Re = 1000. (a) T = 1.0, (b) T = 2.0, (c) T = 3.0

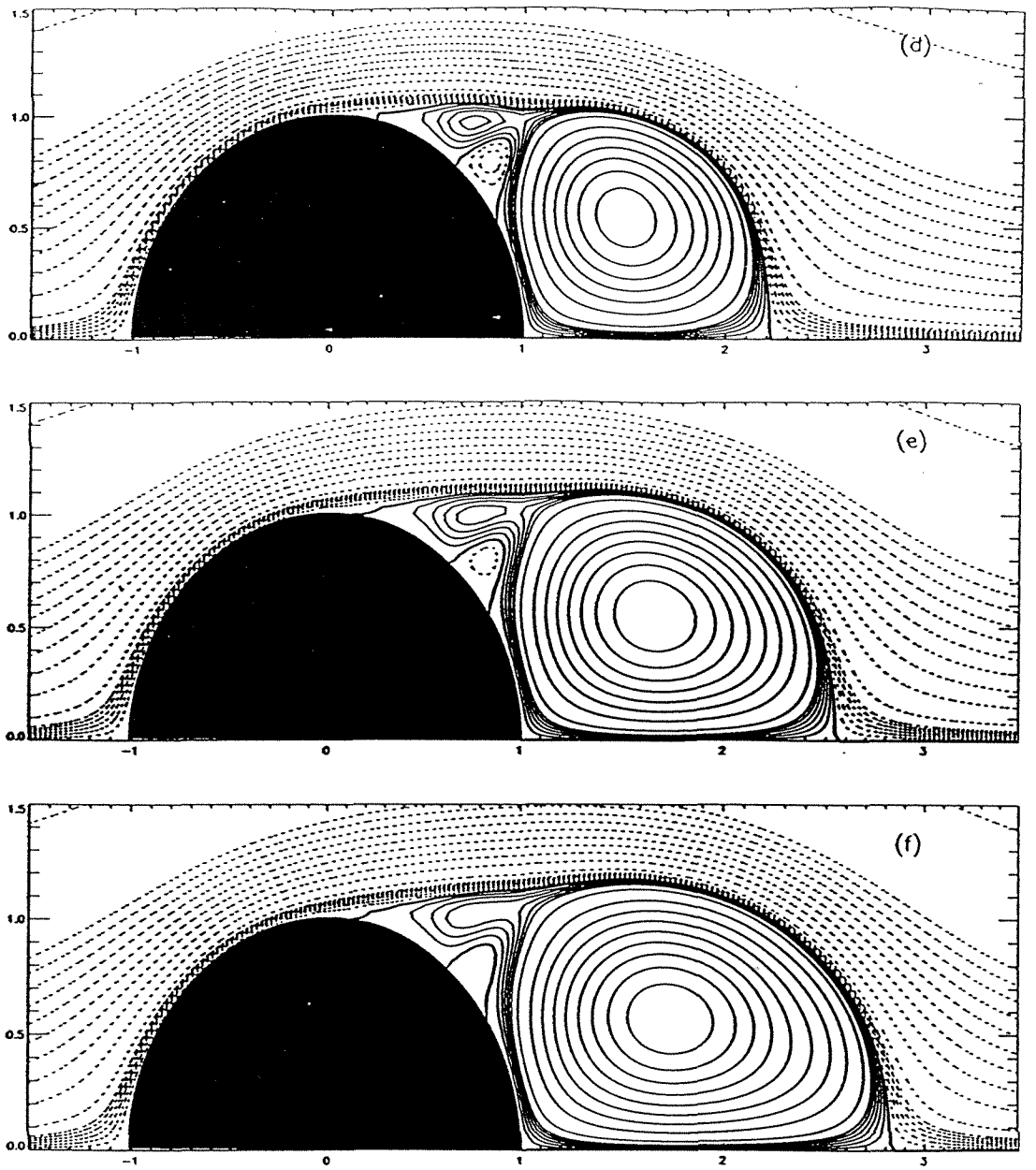


FIG. 7.23b (d) T = 4.0, (e) T = 5.0, (f) T = 6.0. Instantaneous streamlines for an ISCC at Re = 1000

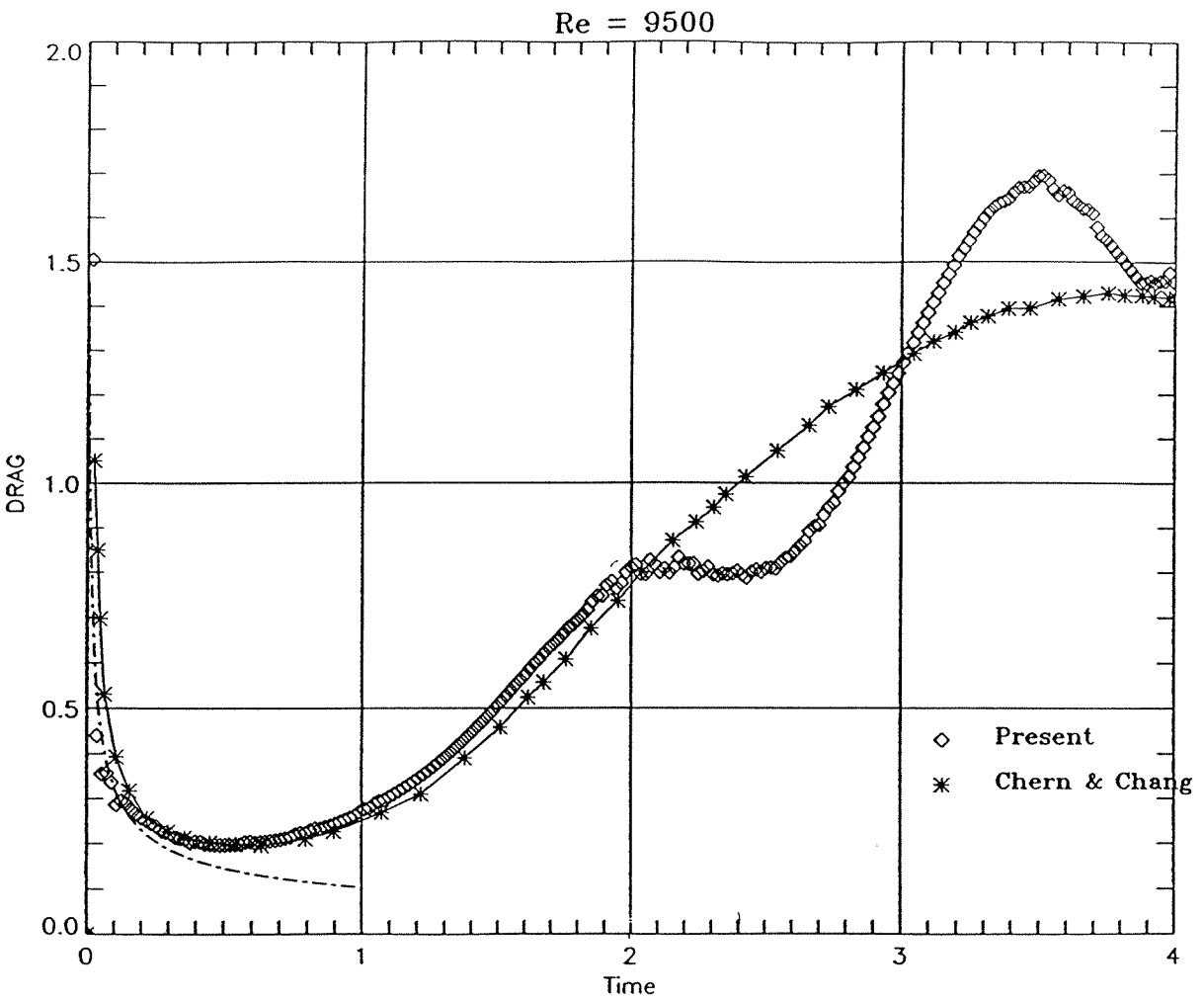
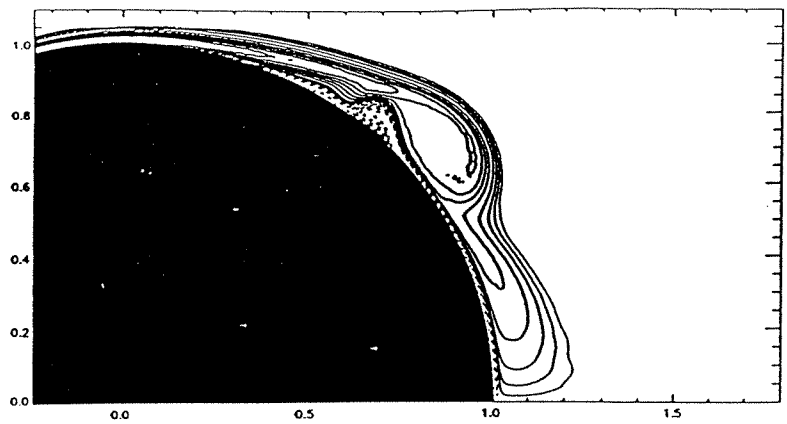
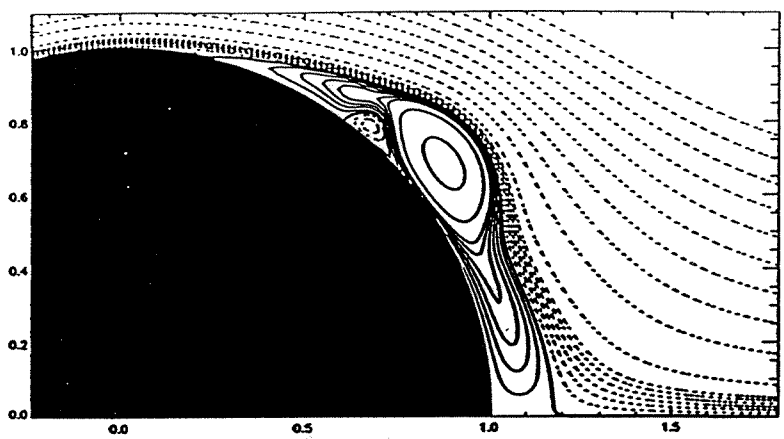


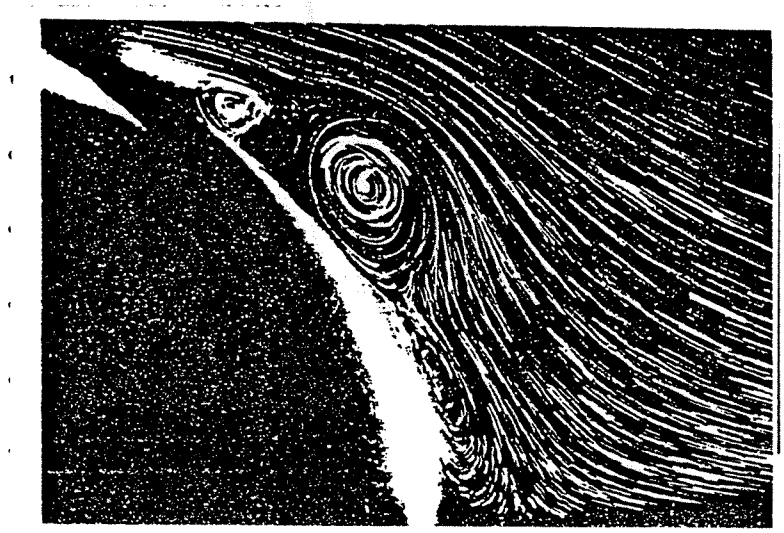
FIG. 7.24 Comparison of the Drag Coefficient of an impulsively started circular cylinder as computed by the present method and that of *CC*. Dashed line: Analytical results of *BY*



(a) Vorticity
(Computations)

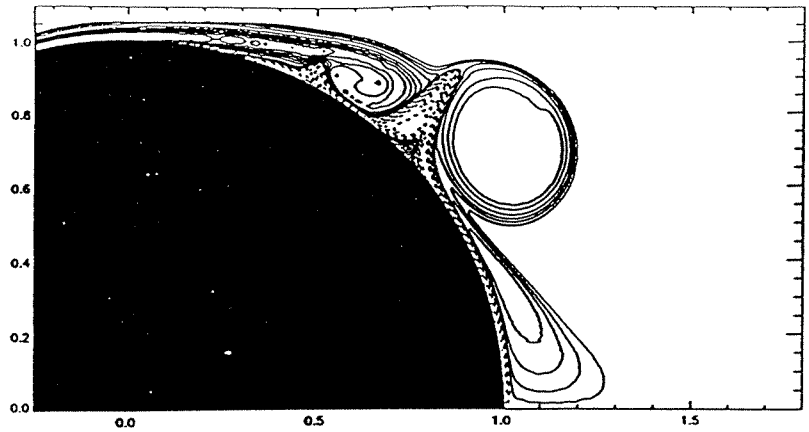


(b) Streamlines
(Computations)

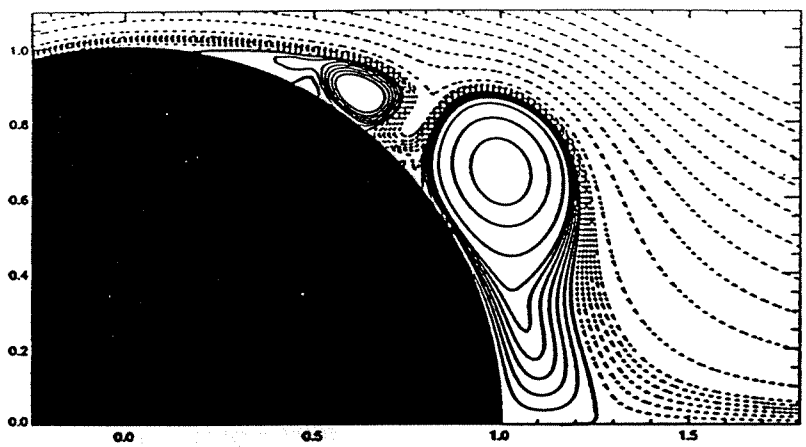


(c) Streaklines
(Experiments)

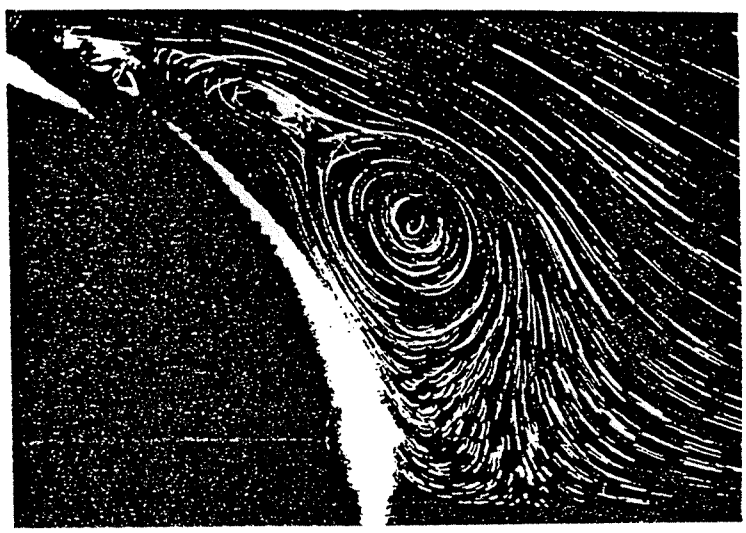
FIG. 7.25 Equivorticity lines (top) and instantaneous streamlines from computations (middle) and experiments (BC) (bottom) for Re = 9500 , T = 2.0



(a) Vorticity
(Computations)

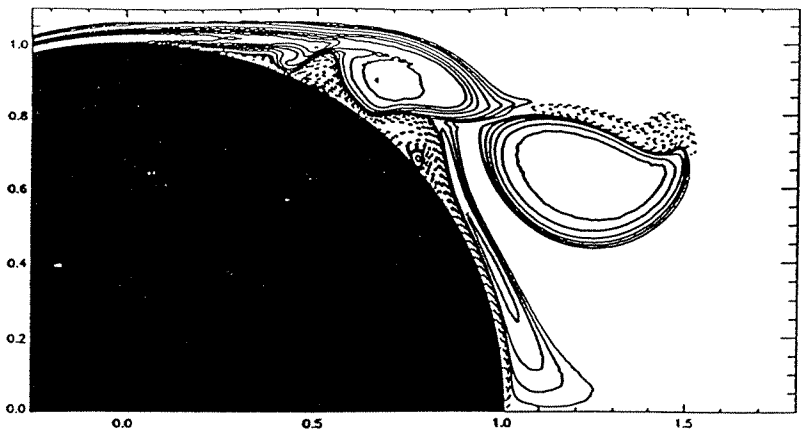


(b) Streamlines
(Computations)

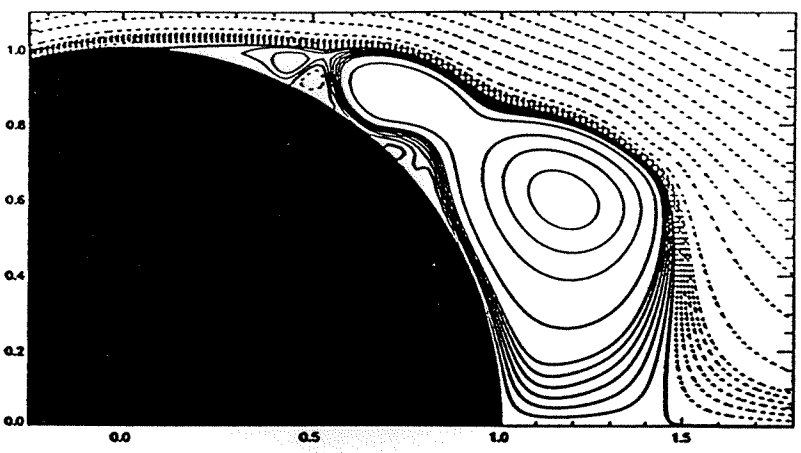


(c) Streaklines
(Experiments)

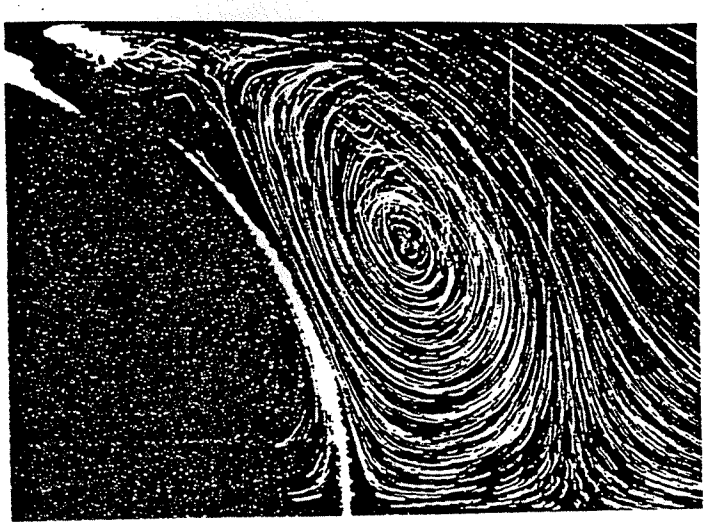
FIG. 7.26 Comparison of computational (present) and experimental results (BC) for Re = 9500, T = 2.5



(a) Vorticity
(Computations)



(b) Streamlines
(Computations)



(c) Streaklines
(Experiments)

FIG. 7.27 Comparison of computational (present) and experimental results (BC) for Re = 9500, T = 3.0.

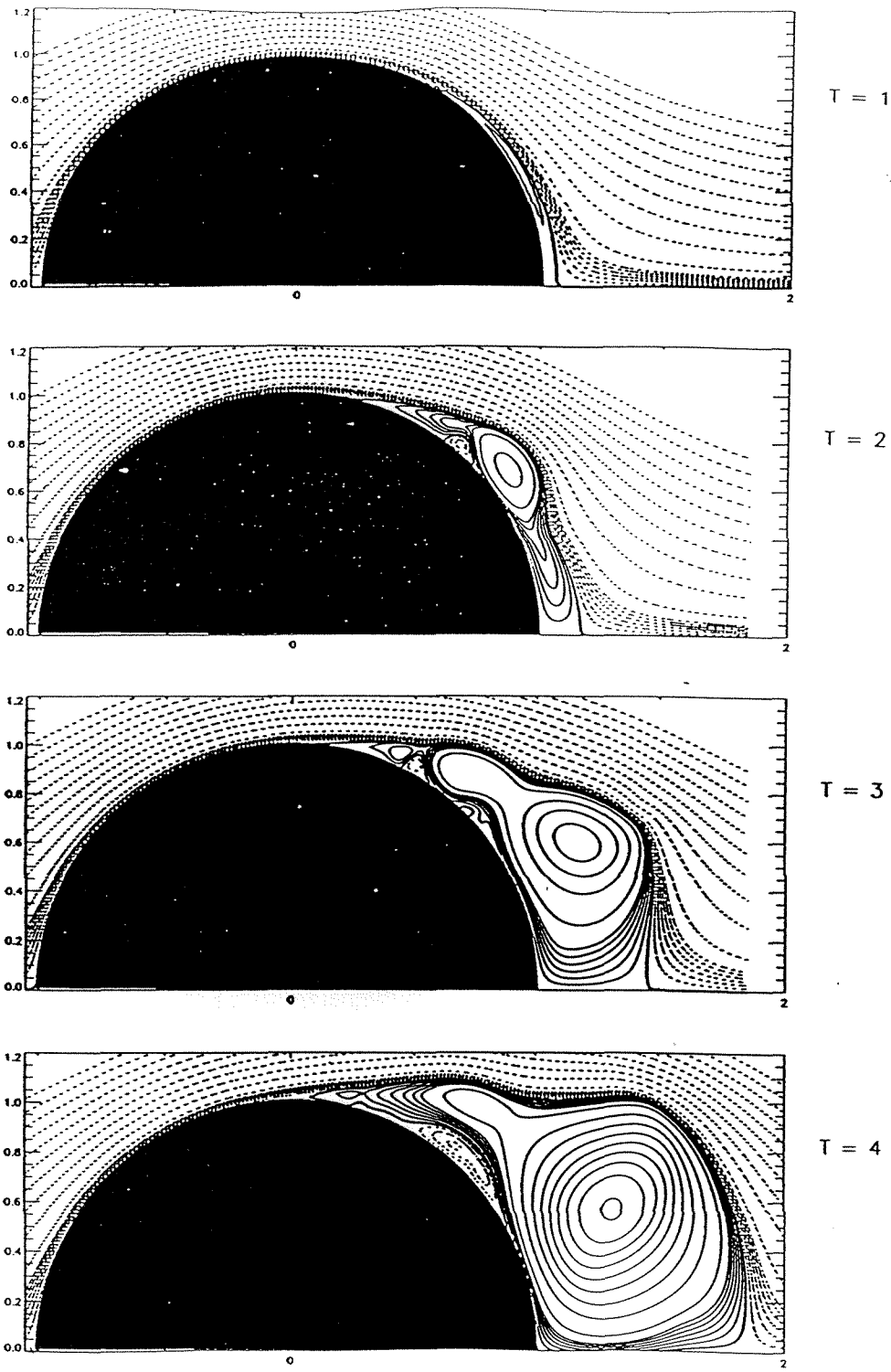


FIG. 7.28 Instantaneous streamlines for an impulsively started circular cylinder at Re = 9500

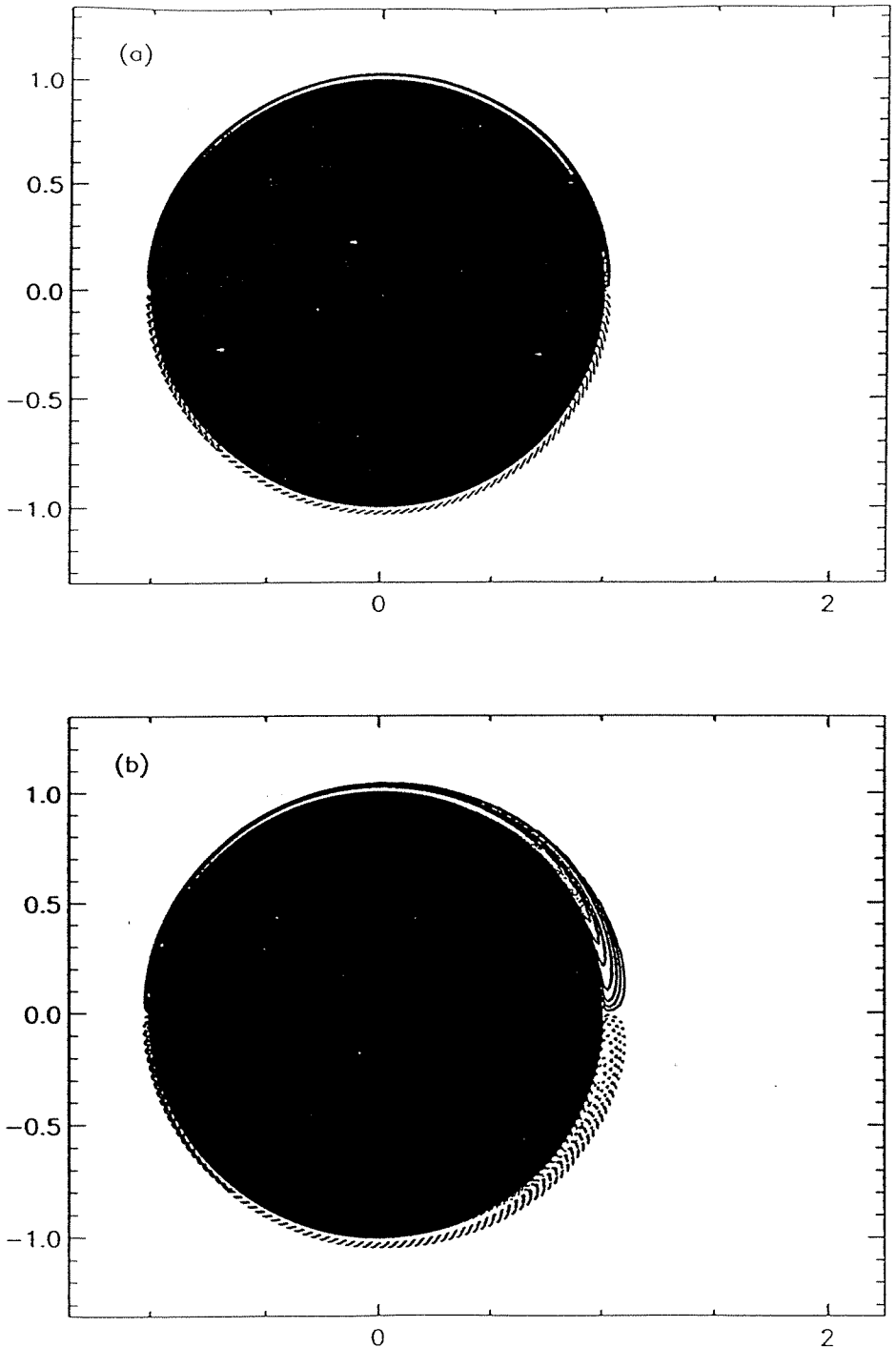


FIG. 7.29a Equivorticity lines of an impulsively started circular cylinder for Re = 9500. (a) T = 0.2, (b) T = 1.0

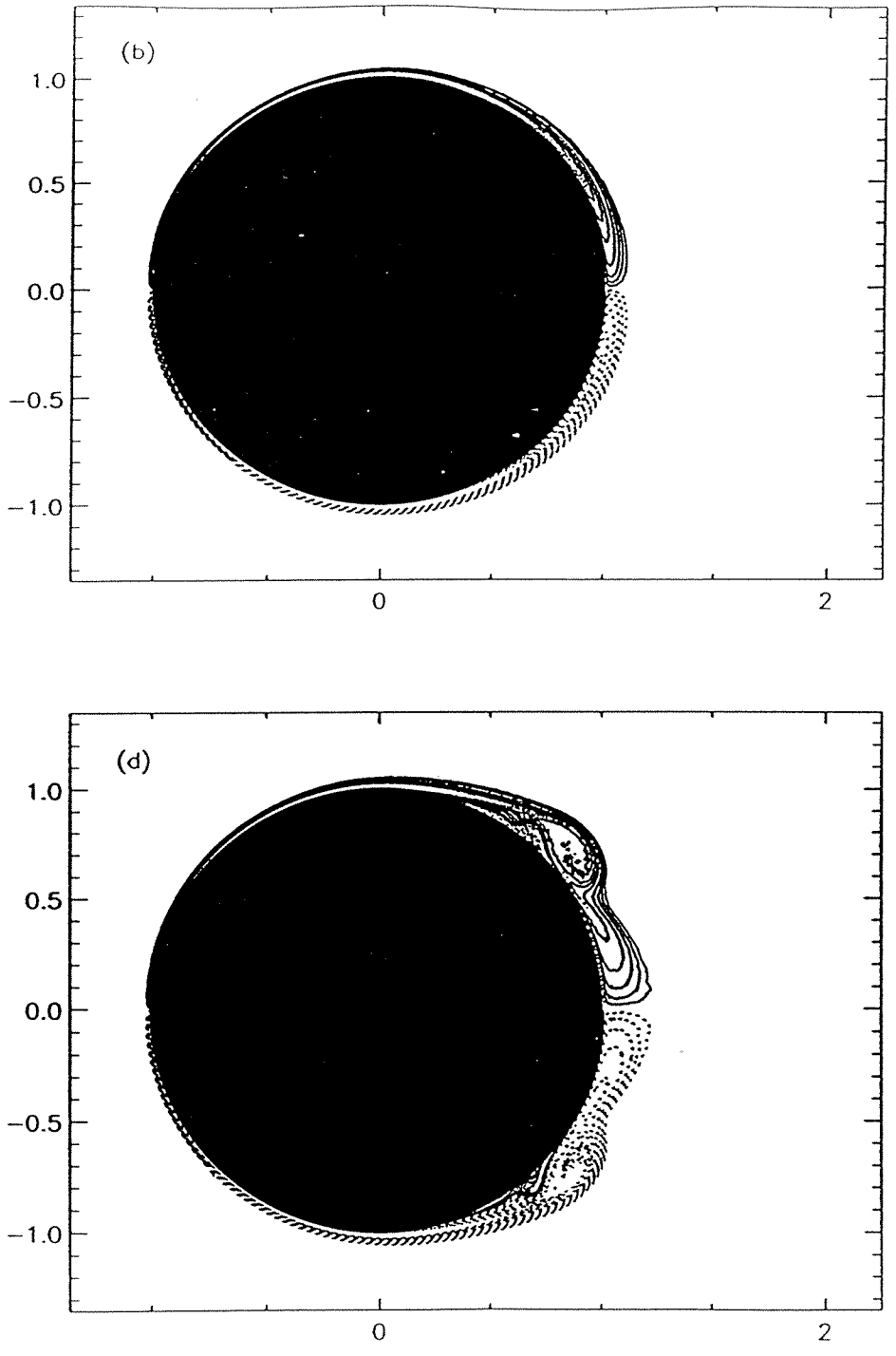


FIG. 7.29b Equivorticity lines of an impulsively started circular cylinder for Re = 9500. (c) T = 1.75, (d) T = 2.0

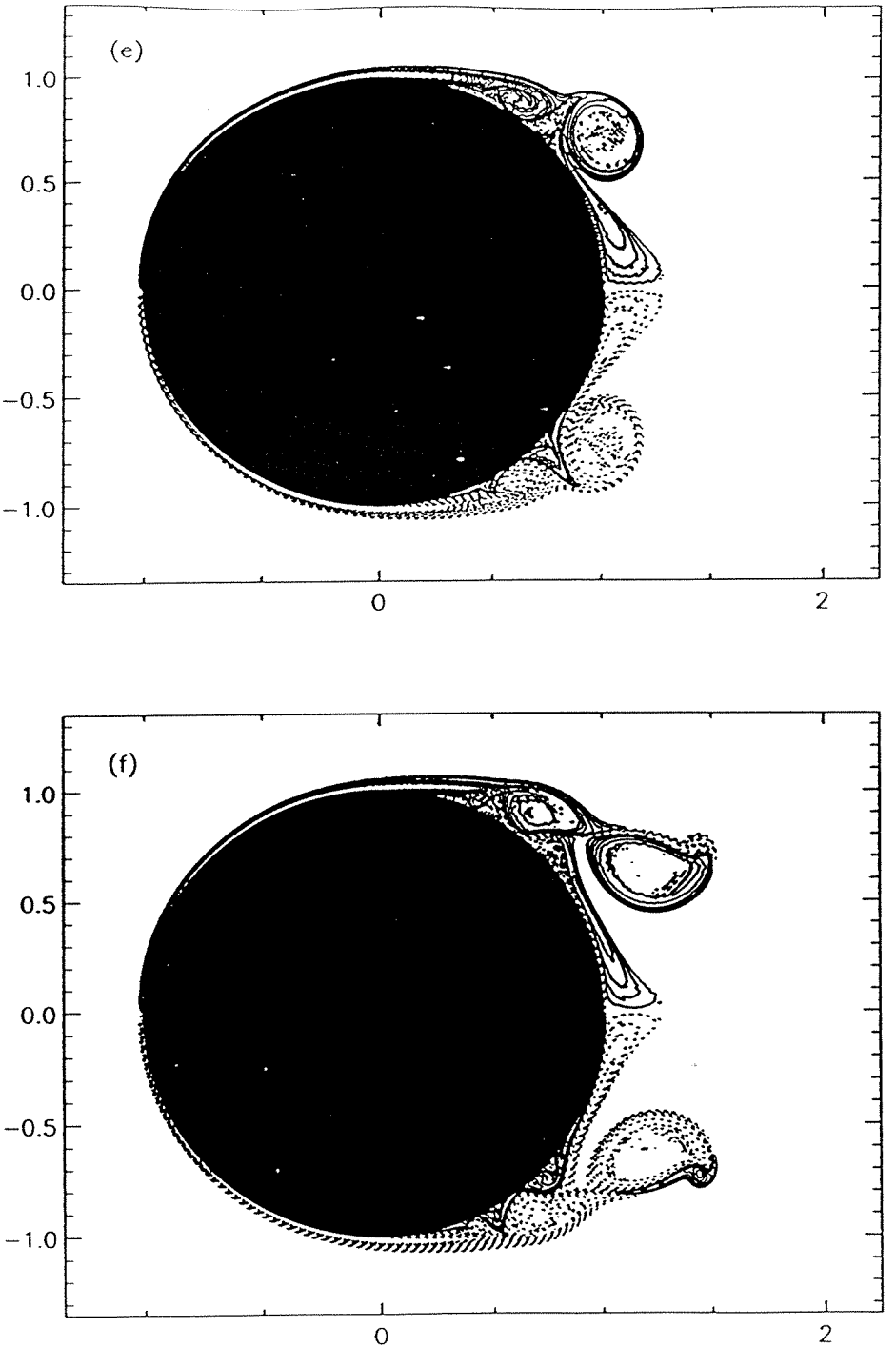


FIG. 7.29c Equivorticity lines of an impulsively started circular cylinder for Re = 9500. (e) T = 2.5, (f) T = 3.0

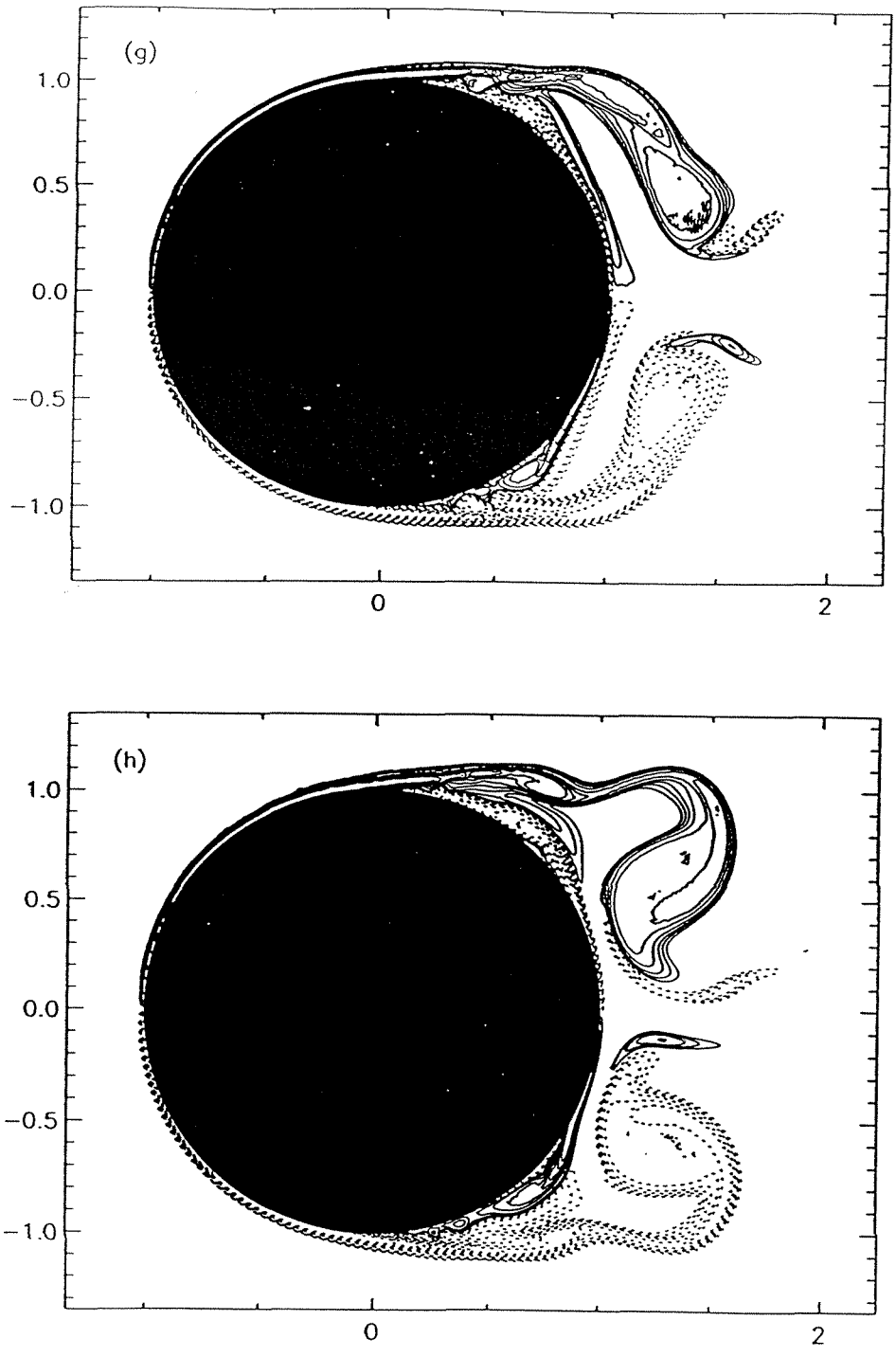


FIG. 7.29d Equivorticity lines of an impulsively started circular cylinder for Re = 9500. (g) T = 3.5, (h) T = 4.0

FIG. 7.30 Vorticity field of an impulsively started circular cylinder at Re = 9500. (T = 1.0, T = 2.0, T = 2.5, T = 3.0, T = 3.5, T= 4.0 : Frames are from top to bottom and left to right).

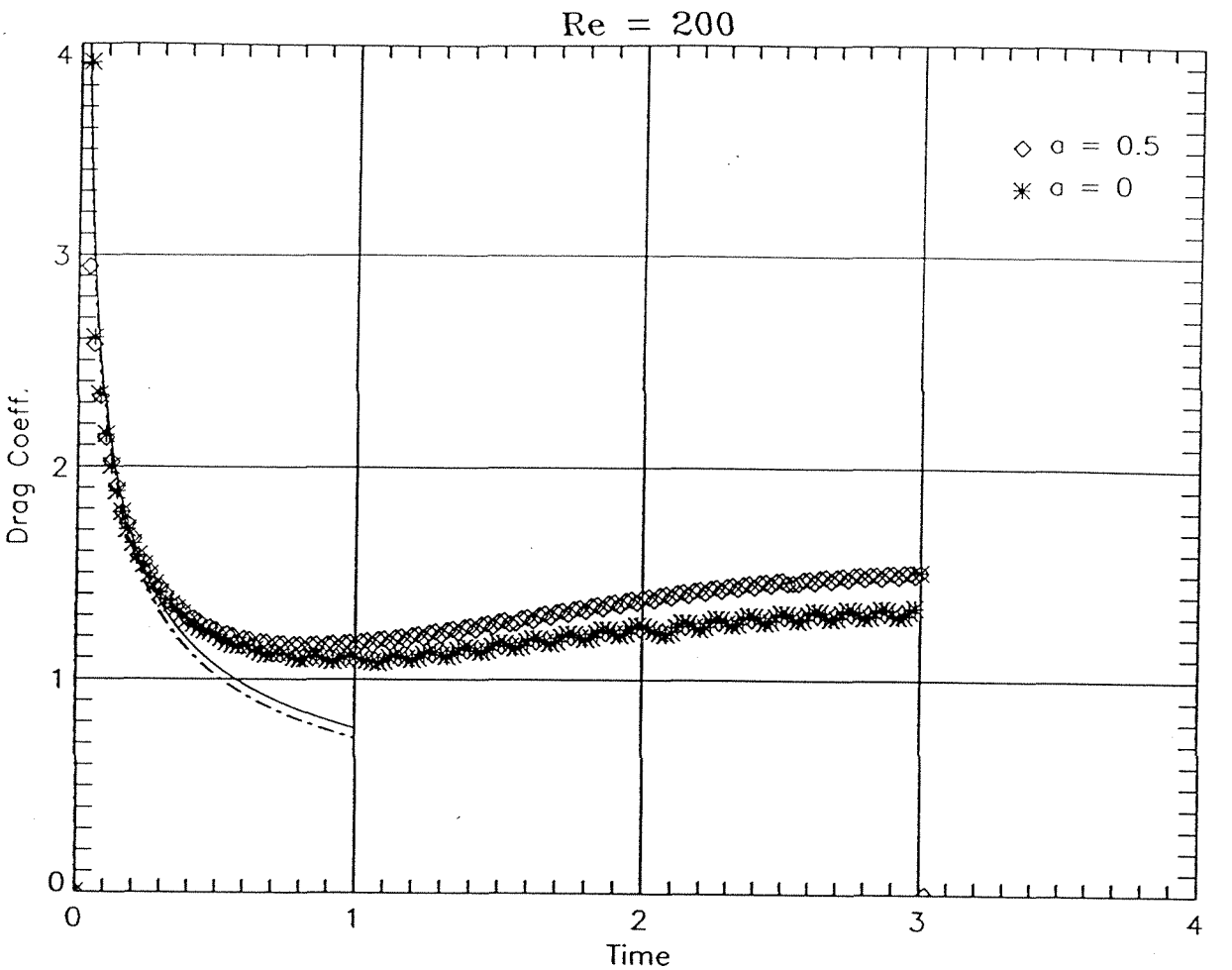


FIG. 7.31 Drag coefficient for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$. Solid line (BY), dashed line (CD).

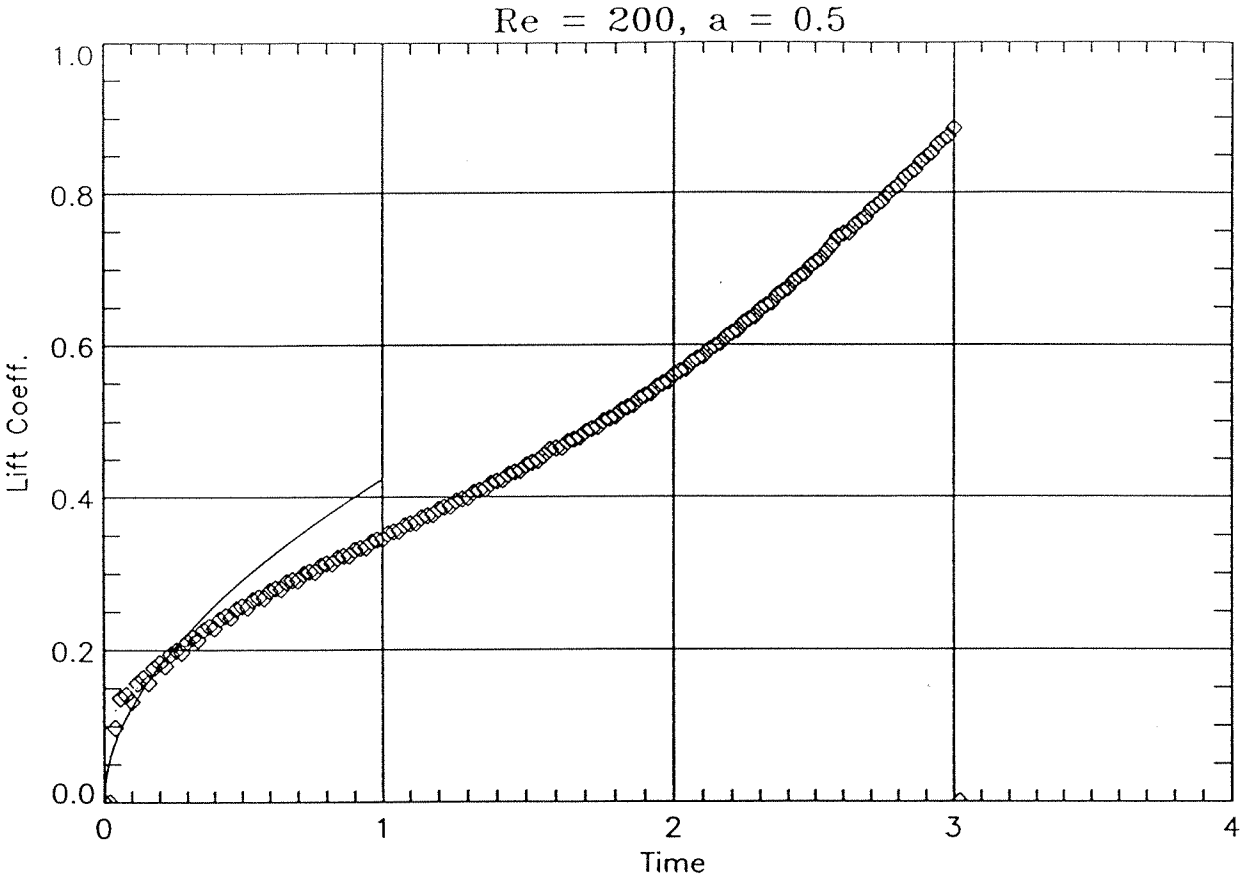


FIG. 7.32 Lift coefficient for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$. Solid Line (BY).

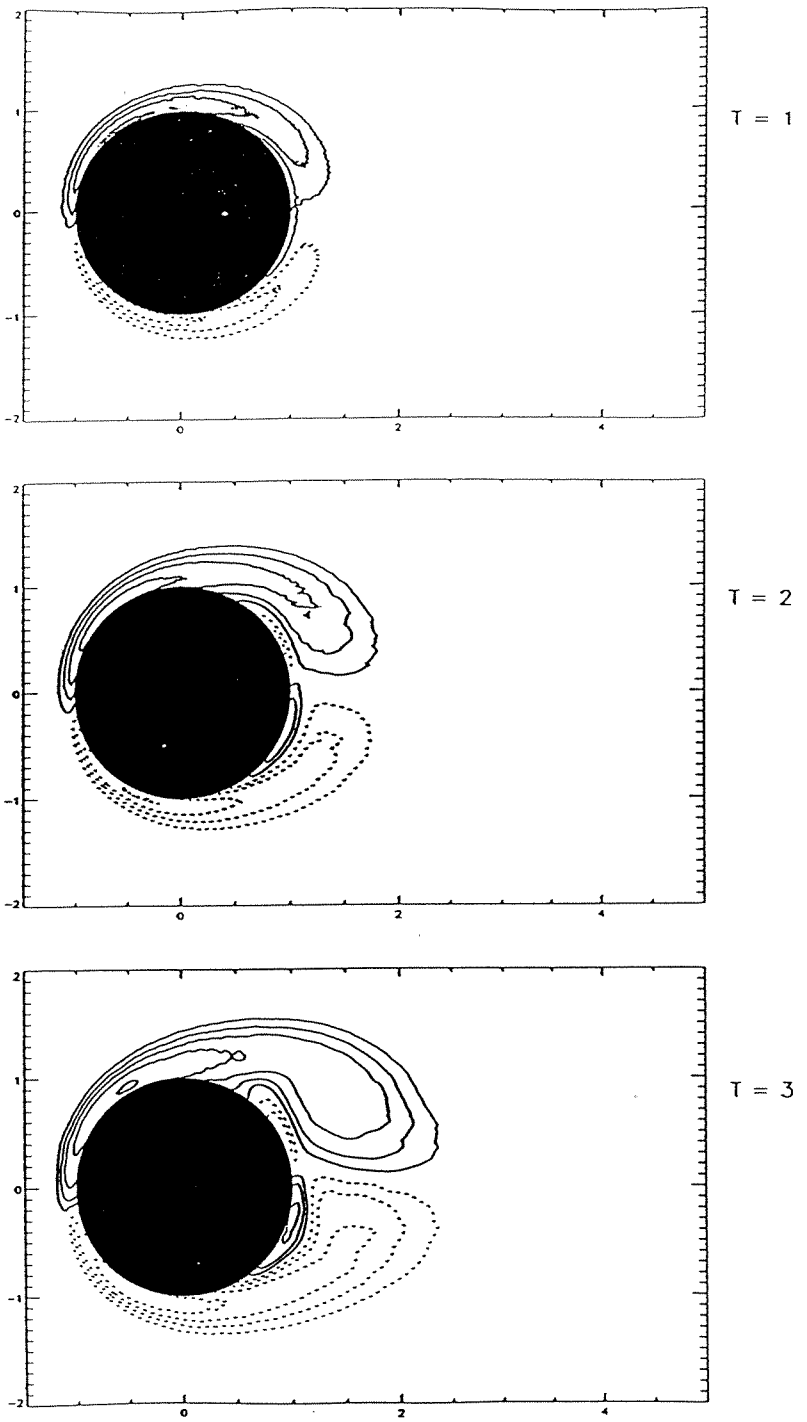


FIG. 7.33 Equivorticity contours for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$

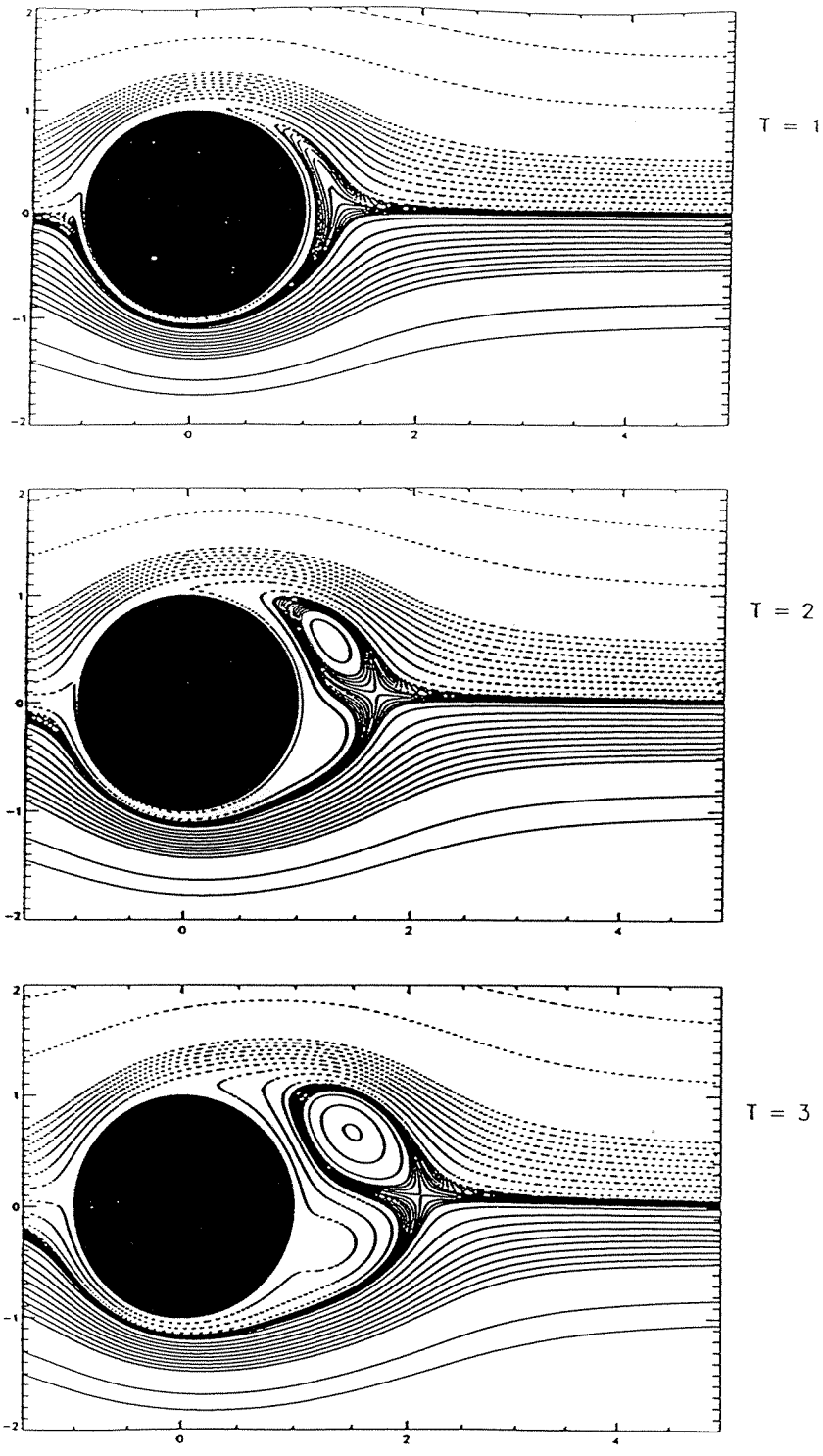


FIG. 7.34 Computed instantaneous streamlines for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$

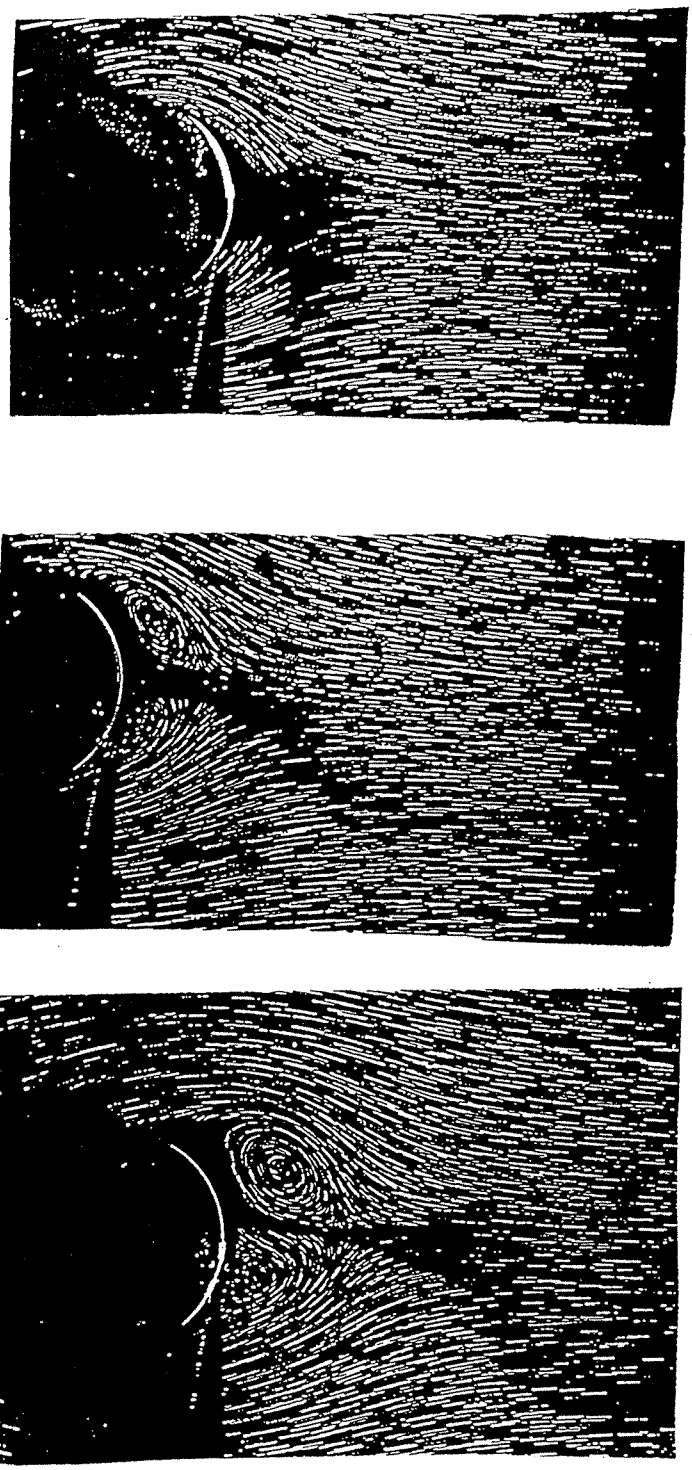


FIG. 7.35 Experimental (from *CM*) instantaneous streamlines for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$

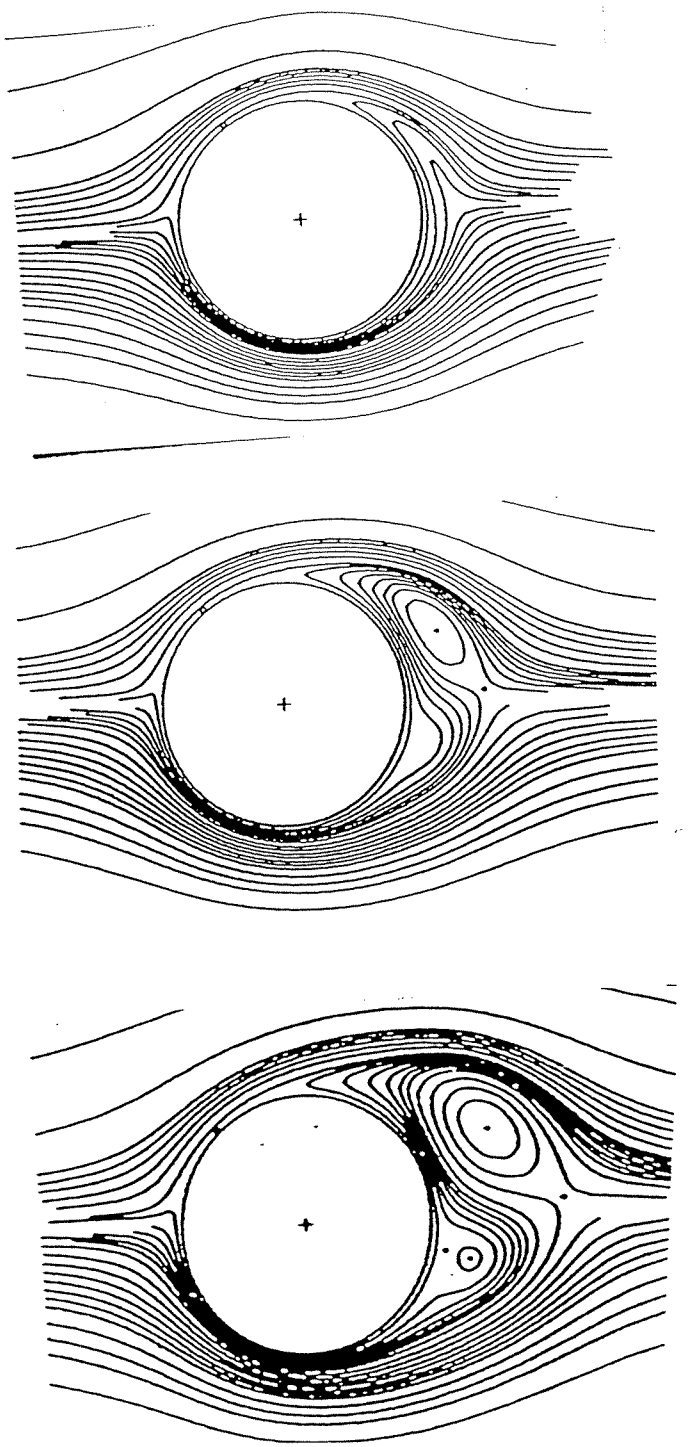


FIG. 7.36 Computational (from *BD*) instantaneous streamlines for an impulsively started translating and rotating cylinder. Re = 200, $\alpha = 0.5$

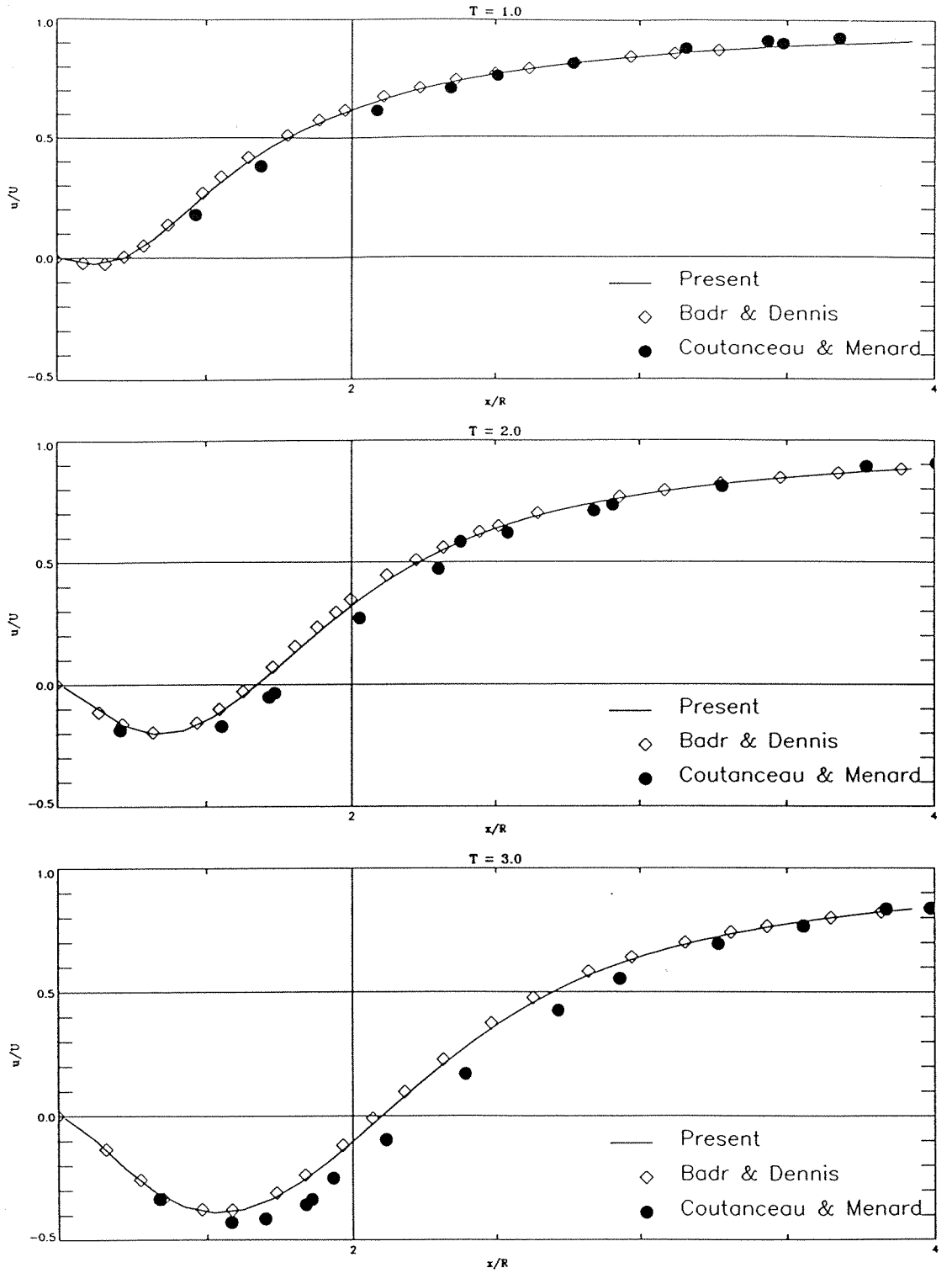


FIG. 7.37 Temporal evolution of the u-velocity component along the x^+ axis for Re = 200, $\alpha = 0.5$. Symbols: experimental data by *CM* and computations by *BD*

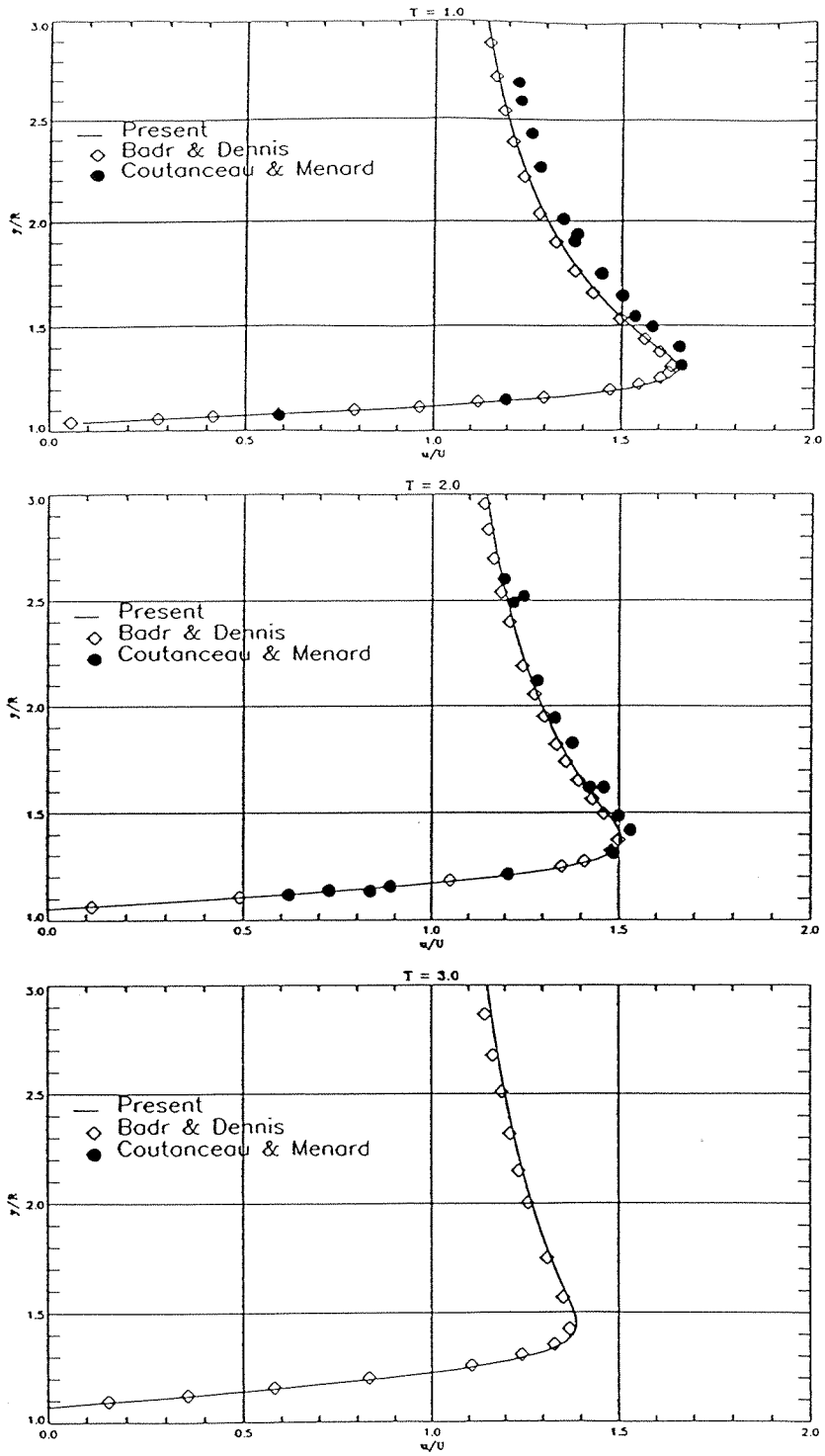


FIG. 7.38 Temporal evolution of the u-velocity component along the y^+ axis for Re = 200, $\alpha = 0.5$. Symbols: experimental data by *CM* and computations by *BD*.

CHAPTER 8

Summary and Conclusions

A Lagrangian numerical method, based on vortex methods, has been developed and implemented for the direct numerical simulations of viscous incompressible flows. The classical vortex method (Leonard, 1980) has been enhanced to properly account for diffusion effects and vorticity creation at solid boundaries.

Vortex methods have several inherent computational advantages, namely their adaptivity, the explicit exact treatment of the far field boundary condition and the ability to obtain physical insight by directly dealing with the vorticity field. In this thesis we concentrated on the major difficulties of vortex methods, namely their computational cost, the accurate treatment of viscous effects and the distortion of the associated Lagrangian grid. The developed numerical method has retained the inherent advantages of vortex methods and it is free of the impediments mentioned above. The key features of the method may be summarized as follows :

- Of paramount importance in the use of vortex methods is the development of a fast algorithm for the evaluation of the velocity field. In order to maintain the capability of the method to treat flows around complex configurations and to explicitly enforce the far field boundary condition we use the scheme of multipole expansions (Greengard and Rohklin, 1987) to determine the velocities of the particles. The cost of the method is reduced to $\mathcal{O}(\mathrm{N})$ as opposed to the $\mathcal{O}(\mathrm{N}^2)$ computations when employing the classical form of the Biot-Savart law. We developed an accurate and robust algorithm and we were able to employ an $\mathcal{O}(10^6)$ vortices for our simulations. In order to compute the velocity field of 10^6 particles *1 minute* of CPU on the CRAY-YMP is required as opposed to the *24 hours* of CPU that would have been required by the classical scheme.

- A robust scheme to account accurately for viscous effects was developed. We employ the scheme of Particle Strength Exchange (PSE) (Raviart, 1987) and complement it with an algorithm that maintains a minimum level of resolution throughout the computational domain thus permitting the accurate description of diffusion effects. Simulations are possible for extended times by periodically restoring the distorted Lagrangian grid while minimizing additional numerical dissipation.

- A novel technique for the enforcement of the no-slip boundary condition in the context of vortex methods has been developed. We follow Lighthill's (1963) model for vorticity creation on a solid surface. We formulate the problem by replacing the surface of the body with a vortex sheet whose strength is determined by the enforcement of the tangential velocity boundary condition. Using a fractional step algorithm this vortex sheet is subsequently nullified resulting in a vorticity flux on the surface of the body. This vorticity flux is diffusively distributed to the vortices in the domain and modifies their strength so that the no-slip condition is enforced. Unlike other vortex methods *no new particles* are generated on the surface of the body but the strength of the existing ones is modified to model vorticity creation in the fluid due to solid walls. The validity of our approach is validated by comparing its results with known analytic solutions.

The robustness and accuracy of the present scheme is demonstrated through a series of direct numerical simulations of incompressible vortical flows.

- Simulations are undertaken to exhibit the capabilities of the method for extended times. The inviscid evolution of a 2:1 elliptical vorticity distribution is considered. By monitoring diagnostics such as the linear and angular impulse we present detailed simulations of the evolution and demonstrate the ability of the method to resolve a wide range of vorticity scales and describe detailed phenomena such as filamentation. The present scheme provides an especially powerful (if not unique) tool for simulations of inviscid vortical flows.

- We focus on simulations for flows around a circular cylinder impulsively started in translation and/or rotation for a wide range of Reynolds numbers (40 - 9500). For the early development of the flow we compare our results with those obtained by analytical solutions (by Bar-Lev and Yang (1975) and Collins and Dennis (1973)). Our comparisons show that the present scheme is able to capture the

initial ($1/\sqrt{T}$) behaviour of the drag coefficient as predicted by the theoretical works. Comparisons are made for moderate times with the experimental works of Bouard and Coutanceau (1980), for the translating cylinder, and Coutanceau and Ménard (1985), for the rotating and translating cylinder, as well as with a variety of other numerical methods. Our comparisons are based on diagnostics such as instantaneous streamlines, velocity profiles and drag coefficients. Through our computations we consider the relevance of such diagnostics and their importance in assessing the validity of numerical simulations. It it is observed that the present method is in excellent agreement with the experimental and (at least qualitatively) most computational works for Reynolds numbers of up to 550. However as the Reynolds number increases discrepancies appear between the present results and those of other computations. It is believed however that due to the high resolution simulations and the accurate resolution of diffusion effects and boundary conditions the results of the present simulations provide an accurate qualitative and quantitative description of the flow field.

- Based on the vorticity description of the flow we analyse the role of the primary and secondary vortical structures in the development of the early stages of the flow and their relation to the drag experienced by the body. It is observed that the main source of drag is the primary vortex formed behind the cylinder. The role of the secondary vorticity is instrumental however as it acts to reduce or enhance the feeding of the primary vortex from the vorticity created on the surface of the body. The secondary vorticity appears to be more active for higher Reynolds numbers as it acts to rearrange the topology of the flow and subsequently affect the drag experienced by the body. Observations are made as to the role of the numerical diffusion to suppress the activities of the secondary vorticity and evidence is given as to the sensitivity of such simulations to the numerical diffusion. Finally preliminary simulations are presented for lifting flows mainly to exhibit the accuracy of the method and its ability to handle a variety of unsteady separated flows.

Suggestions for Future Investigations

A few observations are in order for future investigations for the numerical method and physical problems tractable by it.

As far as the numerical method is concerned :

- Many of the developments described herein for two-dimensional flows can be readily extended to three dimensional simulations. In particular the present method of enforcing the no-slip boundary condition is easily extended to three dimensions and is well suited for vortex methods as it was shown in this work. On the other hand, the scheme of PSE has already been demonstrated to yield accurate results for unbounded three dimensional simulations (Winckelmans, 1987). This representation of viscous effects when complemented with a proper remeshing procedure and a fast algorithm for the velocity evaluation should yield a powerful tool for numerical simulations of unsteady separated three dimensional flows. Work in this direction is already in progress.

- The use of vortex blobs of same size and shape throughout the computational domain is what truly limits the applicability of the method. It would be very beneficial to devise a scheme that would allow for a fine discretization of the vorticity field near the body but allow larger computational elements to represent distant parts of the flow field thus reducing the number of computational elements required. For the evaluation of the velocity field such a scheme is available (*Hou*, 1992). One may try to extend the techniques developed then to the scheme of PSE thus developing a superior Lagrangian method. Some thought should be given however to the logistics of this approach and the additional difficulty in devising fast algorithms. Alternatively one may consider, for relatively accurate computations, to abandon the requirements for overlap at distances removed from the body thus not increasing the number of elements required to describe the remote flow field. Such an approach however would require extensive and thorough research to determine the parameter space and remove the term 'direct numerical' from the simulations.

A plethora of interesting physical problems may be investigated using the present numerical method

- A more systematic study of the evolution of vortical structures in an infinite domain would be in order. The ability of the scheme to simulate inviscid and viscous flows may be explored to provide us with an understanding of the effects of viscosity on the development of such flows.

- Without major modifications to the existing algorithm more complicated unsteady separated flows may be considered. The cylinder may be forced to undertake rotary oscillations besides translating and steadily rotating and it would be interesting to perform computations for the experiments of Tokumaru and Dimotakis (1991).

- With a few modifications of our code (the formulation is already developed), flows around more complex configurations may be examined. Flows of particular interest would be those behind a flat plate. We may validate previous computational and experimental works and examine the role of the secondary separation for flows where the primary separation points (such as the tips of the plate) are well defined.

- Finally the present algorithm may be used to examine control mechanisms for unsteady separated flows. As its primary variable is the vorticity field, one may easily gain physical insight for the passive development of the flows at hand and devise active control mechanisms for flows around arbitrary configurations.

References

Anderson C. (1986). A Method of Local Corrections for Computing the Velocity Field Due to a Distribution of Vortex Blobs.. *J. Comput. Phys.* **62**, 111-123.

Appel A. (1985). An Efficient Program for Many-Body Simulation. *SIAM J. Sci. Stat. Comput.* **6**, 85.

Badr H.M. and Dennis S.C.R. (1985). Time-dependent Viscous Flow Past an Impulsively Started Rotating and Translating Circular Cylinder. *J. Fluid Mech.* **158**, 447-488.

Baker G.R. and Shelley M.J. (1986). Boundary Integral Techniques for Multi- connected Domains. *J. Comput. Phys.* **64**, 112-132.

Bar-Lev M. and Yang H. T. 1975. Initial Flow Field Over an Impulsively Started Circular Cylinder. *J. Fluid Mech.* **72**, 625-647.

Barnes J.E. (1990). A Modified Tree Code: Don't Laugh; It Runs. *J. Comput. Phys.* **87**, 148-60.

Barnes J.E. and Hut P. (1986). A Hierarchical $\mathcal{O}$(N logN) Force-Calculation Algorithm. *Nature* **324**, 446-449.

Beale J.T. (1986). On the Accuracy of Vortex Methods at Large Times. *Proc.Workshop on Comp. Fluid Dyn. and React. Gas Flows, I.M.A., Univ.of Minnesota.*

Blasius H. (1908). Grenzchichten in Flüssigkeiten mit Kleiner Reibug.. *Z. Angew. Math. Phys. (Engl. transl., NACA TM-1256* **56**, 1.

Bouard R. and Coutanceau M. (1980).. The Early Stage of Development of the Wake Behind an Impulsively Started Cylinder for 40 $\leq$Re$\leq$ 10^4. *J. Fluid Mech.* **101**, 583-607.

Chang C.C. and Chern R.L. (1991). A Numerical Study of Flow Around an Impulsively Started Circular Cylinder by a Deterministic Vortex Method. *J. Fluid Mech.* **233**, 243-263.

Chen Y.M., Ou Y.R. and Pearlstein A.J. (1991). Development of the Wake Behind a Circular Cylinder Impulsively Started into Rotatory and Rectilinear Motion: Intermediate Rotation Rates. *ICASE Rep. No. 91-10.*

Chorin A.J. (1973). Numerical Study of Slightly Viscous Flow. *J. Fluid Mech.* **57**, 380-392.

Christiansen J. P. (1973). Vortex Methods for Flow Simulation. *J. Comput. Phys.* **13**, 363.

Collins W.M. and Dennis S.C.R. (1973). Flow Past an Impulsively Started Circular Cylinder. *J. Fluid Mech.* **60**, 105-127.

Cottet S.H (1987). *Analyse Numérique des Méthodes Particulaires pour Certains Problèmes Non-Linéaires.* Thèse d' état, Université Pierre et Marie Curie.

Cottet G.H. (1990). A Particle-Grid Superposition Method for the Navier-Stokes Equations. *J. Comput. Phys.* **89**, 301-318.

Coutanceau M. and Ménard C. (1985). Influence of rotation on the Near Wake Development Behind an Impulsively Started Circular Cylinder. *J. Fluid Mech.* **158**, 399-446.

Degond P. and Mas-Gallic S. (1989). The weighted Particle Method for Convection - Diffusion Equations, Part I : The Case of an Isotropic Viscosity, Part II : The Anisotropic Case.. *Math. Comput.* **53**, 485-526.

Dritschel D.G. (1989). Contour Dynamics and Contour Surgery: Numerical Algorithms for Extended, High-Resolution Modelling of Vortex Dynamics in Two Dimensional, Inviscid, Incompressible Flows.. *Comput. Physics Rep.* **10**, 77-146.

Dukowicz J.K. and Kodis J.W. (1987). Accurate Conservative Remapping(Rezoning) for Arbitrary Lagrangian-Eulerian Computations. *SIAM J. Sci. Stat. Comput.* **8**, 305-321.

Farmer C.L. (1985). A Moving Point Method for Arbitrary Peclet Number Multidimensional Convection-Diffusion Equations.. *IMA J, Numer. Anal.* **5**, 465-480.

Fishelov D. (1990). A New Vortex Scheme for Viscous Flows. *J. Comput. Phys.* **86**, 211-224.

Friedmann A.. *Partial Differential Equations of Parabolic Type.* Prentice-Hall Inc., Englewood Cliffs, N.J. 1964.

Fritts M.J., Crowley W.P. and Trease H.E. eds.. *The Free Lagrange method.* in Lecture Notes in Physics 238, Springer-Verlag. Berlin, New York, 1985

Gray A. and Mathews G.B. . *A Treatise on Bessel Functions and Their Applications to Physics.* MacMillan and Co., London, 1952.

Greengard C. (1985). The Core Spreading Vortex Method Approximates the Wrong Equation. *J. Comput. Phys.* **61**, 345-348.

Greengard L. and Rokhlin V. (1987). A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325-348.

Greengard L. and Strain J. (1990). A Fast Algorithm for the Evaluation of Heat Potentials. *Commun. Pure Appl. Math.* **XLIII**, 949-963.

Haddard B. ,Clausset F. and Combes F. (1991). Vectorising the Smooth Particle Hydrodynamics. *J. Comp. Physics* **97**, 103-126.

Hald H.O. (1979). Convergence of Vortex Methods for Euler's Equations. II. *SIAM J. Numer. Anal.* **16**, 726-755.

Hernquist L. (1990). Vectorization of Tree Traversals. *J. Comput. Phys.* **87**, 137-147.

Hess J.L. (1990). Panel Methods in Computational Fluid Dynamics. *Ann. Rev. Fluid Mech.* **22**, 255-274.

Hess J.L. (1975). Review of Integral-Equation Techniques for Solving Potential Flow Problems with Emphasis on the Surface-Source Method. *Comp. Methods in Applied Mech. and Eng.* **5**, 145-196.

Hockney R.W. and Eastwood J.W. . *Computer Simulation Using Particles.* McGraw-Hill Inc., 1981.

Hou T.Y. (1990). Convergence of a Variable Blob Vortex Method for the Euler and Navier-Stokes Equations. *SIAM J, Numer. Anal.* **27**, 1387-1404.

Hou T.Y. and Wetton B.T.R. (1992). Convergence of a Finite Difference Scheme for the Navier-Stokes Equations. *SIAM J, Numer. Anal.* **29**, 615-639.

Huberson S. (1986). *Modélisation asymptotique et numérique de noyaux tourbillonaires enroulés.* Thèse d' état, Université Pierre et Marie Curie.

Hung S.C. and Kinney R.B. (1988). Unsteady Viscous Flow Over a Grooved Wall: A Comparison of Two Numerical Methods. *Intern. J. for Numer. Meth. in Fluids* **8**, 1403-1437.

Isaacson E. and Keller H.B.. *Analysis of Numerical Methods.* John Wiley and Sons, 1966.

Jolles A.and Huberson S. (1989). Numerical Simulation of Navier-Stokes Equations by Mean of Particle-Mesh Methods. *Notes et documents LIMSI: 89-4, L.I.M.S.I., C.N.R.S..*

Kambe T. (1983). A Class of Exact Solutions of Two-Dimensional Viscous Flow. *J. Phys. Soc. Japan.* **52**, 834-841.

Katzenelson J. (1989). Computational Structure of the N-Body Problem. *SIAM J. Sci. Stat. Comput.* **11**, 787-815.

Kinney R.B. and Paolino M.A. (1974). Flow Transient near the Leading Edge of a Flat Plate Moving Through a Viscous Fluid. *ASME J. Appl. Mech.* **41(4)**, 919-924.

Kinney R.B. and Cielak Z.M. (1977). Analysis of Unsteady Viscous Flow Past an Airfoil: Part I - Theoretical Development. *AIAA J.* **15 (12)**, 1712-1717.

Lecointe Y. and Piquet J. (1984). On the Use of Several Compact Methods for the Study of the Incompressible Viscous Flow Around a Circular Cylinder. *Comp. and Fluids* **12**, 255-280.

Leonard A. (1980). Vortex Methods for Flow Simulation. *J. Comput. Phys.* **37**, 289-335.

Leonard A. (1985). Computing Three-dimensional Incompressible Flows with Vortex Elements. *Ann. Rev. Fluid Mech.* **17**, 523-559.

Lighthill M.J. (1963). *Introduction. Boundary Layer Theory, J. Rosenhead Ed.*. Oxford Univ. Press, New York 1963, pp.54-61.

Lucquin-Desreux B and Mas-Galic S. (1990). A Conservative Particle Approximation for a Boundary Advection-Diffusion Problem. *R90022, Universite Pierre et Marie Curie, C.N.R.S.*.

Martensen E. (1959). Berechnung der Druckverteilung an Gitterprofilen in ebener Potentialströmung mit einer Fredhölmschen Integralgleichung. *Archive for Rational Mechanics and Analysis* **3**, 235-270.

McIntyre A. (1986). Boundary Integral Solutions of the Heat Equation. *Math. Comp.* **46**, 71-79.

Mas-Gallic S. (1987). *Contribution à l' Analyse Numérique des Méthodes Particulaires.* Thèse d' état, Université Pierre et Marie Curie.

Mas-Gallic S. (1990). Une Methode Particulaire Deterministe Incluant Diffusion et Conditions aux Limites. *C.R.Sci. Paris 310* **Serie I**, 465-468.

Meiburg E. (1989). Incorporation and Test of Diffusion and Strain Effects in the Two-Dimensional Vortex Blob Technique. *J. Comput. Phys.* **82**, 85-93.

Melander M.V., McWilliams J.C. and Zabusky N.J. (1987). Axisymmetrization and Vorticity Gradient Intensification of an Isolated Two Dimensional Vortex Through Filamentation. *J. Fluid Mech.* **178**, 137-159.

Mikhlin S.G.. *Integral Equations and Their Applications to Certain Problems of Mechanics, Mathematical Physics and Engineering.* 2nd ed., Macmillan, New York, 1964.

Milizzano F. and Saffman P.G. (1977). The Calculation of Large Reynolds Number Two-Dimensional Flow Using Discrete Vortices with Random Walk. *J. Comput. Phys.* **23**, 380-392.

Makino J. (1990). Vectorization of a Treecode. *J. Comput. Phys.* **87**, 148-60.

Monaghan J.J. (1985). Particle Methods for Hydrodynamics. *Comp. Phys. Reports* **3**, 71-124.

Morchoisne Y., Le T.H. and Ryan J. (1987). Fourier Condition in Integral Methods: Application to Thin Bodies. *Rech. Aero* 4, 71-74.

Payne R.B. (1958). Calculation of Viscous Unsteady Flows Past a Circular Cylinder. *J. Fluid Mech.* **4**, 81-86.

Pépin F. (1990). *Simulation of the Flow Past an Impulsively Started Cylinder Using a Discrete Vortex Method.* Ph. D. thesis, Caltech.

Pepin F. and Leonard A. (1990). Concurrent Implementation of a Fast Vortex Method. *5th Distributed Memory Computing Conference* **I**, 453-462.

Prager W. (1928). Die Druckverteilung an Körpern in ebener Potentialströmung. *Physik. Zeitschr.* **29**, 865-869.

Prandtl W. (1925). The Magnus Effect and Windpowered Ships. *Wissenschaften* **13**, 93-108.

Pullin D.I. (1992). Contour Dynamics Methods. *Ann. Rev. Fluid Mech.* **24**, 89-115.

Raviart P.-A. (1987). *Méthodes particulaires.* Lecture notes, Ecole d'été d'analyse numérique, Centre d'étude du Bréau-sans-nappe, France.

Roache P.T.. *Computational Fluid Dynamics.* Hermosa Publishers, Albuquerque, 1972.

Rosenhead L. (1931). The Formation of Vortices from a Surface of Discontinuity. *Proc. Roy. Soc., Ser. A* **134**, 170.

Salmon J.K. (1991). *Parallel Hierarchical N-Body Methods.* Ph. D. thesis, Caltech.

Sarpkaya T. (1989). Computational Methods with Vortices. *J. Fluids Engng.* **111**, 5-52.

Schmall R.A. and Kinney R.B. (1974). Numerical Study of Unsteady Viscous Flow Past a Lifting Plate. *AIAA J.* **12**, 1566-1573.

Schoenberg I.J.. *Cardinal Spline Interpolation.* SIAM, Philadelphia, 1973.

Smith P.A. and Stansby P.K. (1988). Impulsively Started Flow Around a Circular Cylinder by the Vortex Method. *J. Fluid Mech.* **194**, 45-77.

Ta Phuoc Loc (1980). Numerical Analysis of Unsteady Secondary Vortices Generated by an Impulsively Started Circular Cylinder. *J. Fluid Mech.* **100**, 111-128.

Ta Phuoc Loc and Bouard R. (1985). Numerical Solution of the Early Stage of the Unsteady Viscous Flow Around a Circular Cylinder: a Comparison with Experimental Visualization and Measurements. *J. Fluid Mech.* **160**, 93-117.

Taslim M.E. Kinney R.B. and Paolino M.A. (1984). Analysis of Two-Dimensional Viscous Flow over Cylinders in Unsteady Motion. *AIAA J.* **22**, 586-594.

Tokumaru P. and Dimotakis P. (1991). Rotary Oscillation Control of a Cylinder Wake. *J. Fluid Mech.* **224**, 77-90.

Tritton D.J. (1959). Experiments on the Flow Past a Circular Cylinder at Low Reynolds Numbers. *J. Fluid Mech.* **6**, 547-567.

Van Dommelen L. and Rundesteiner E.A. (1989). Fast, Adaptive Summation of Point Forces in the Two-Dimensional Poisson Equation. *J. Comput. Phys.* **83**, 126-148.

Wang X. and Dalton C. (1991). Numerical Solutions for Impulsively Started and Decelerated Viscous Flow Past a Circular Cylinder. *Int. J. for Numer. Meth. in Fluids* **12**, 383-400.

Winckelmans G. 1989. *Topics in vortex methods for the computation of three- and two-dimensional incompressible unsteady flows.* Ph. D. thesis, Caltech.

Wu J.C. and Thompson J.F. (1973). Numerical Solution of Time-Dependent Incompressible Navier-Stokes Equations Using an Integro-Differential Formulation. *Computers and Fluids* **1**, 197-215.

Wu J.C. (1976). Numerical Boundary Conditions for Viscous Flow Problems. *AIAA J.* **14**, 1042-1049.

Zabusky N.J., Hughes M.H. and Roberts K.V. (1979). Contour Dynamics for the Euler Equations in Two-Dimensions. *J. Comput. Phys.* **30**, 96-106.