Simulation of the Flow Past an Impulsively Started Cylinder Using a Discrete Vortex Method

Thesis by
François Michel Pépin

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

California Institute of Technology
Pasadena, California
1990
(Submitted May 11th 1990)

to

Sonia Pelletier

Geneviève Bergeron

Anne-Marie Edwards

Nathalie Croteau

Barbara Daigneault

Maud Haviernick

Annie Turcotte

Maryse Leclair

Barbara Maria Klueznick

Maryse Laganière

Anne-Marie Lemay

Michèle Richard

Hélène Colgan

Annie Saint-Arneault

senselessly murdered on December 6th 1989. Six months after, I am still fighting tears as I write these lines. Time has yet to soothe our sorrow and we will make sure that it never completely does. We will never forget. On ne vous oubliera jamais

Acknowledgments

This is by far the most pleasant section to write and not only because I kept it for last. It is the point where, at the end of a long process, one can look back and thank the people whose contributions were instrumental in the success of that process.

First and foremost, I would like to express my gratitude to Professor Anthony Leonard for his guidance, his patience, and especially, for letting his students follow their own course, which is rarely the shortest but in the end, invariably turns out to be the most gratifying one.

This study would not have been possible without the financial support of fonds F.C.A.R., NSERC, AFOSR, and the Department of Energy; thanks to the Québécois, Canadian and American taxpayers respectively. I also wish to thank Heidi Lorenz-Wirzba, the hypercube maestro, for demystifying the "system," Eric Van de Velde for many enlightening discussions on the virtues of concurrent computing, Madeleine Coutanceau for the permission to reproduce Bouard & Coutanceau's flow visualizations, and Dr. Paul Messina for granting access to the Caltech Concurrent Supercomputing Facilities.

Thanks to Riccardo Bonazza and Harry Hamaguchi for sharing their photographic expertise and to Professor Paul Dimotakis for providing the graphic software and the TEX macros which have made the redaction of this document almost a pleasure. This thesis also greatly benefited from the suggestions of a tremendous proofreading team: Petros Koumoutsakos, Grégoire Winckelmans, Martin Brouillette, and Anthony Leonard; thanks to all of them for their sharp eyes and healthy skepticism.

I am grateful to all the friends who have made the GALCIT years enjoyable. Also, nominated in the same category, G. Trudeau, B. Breathed and B. Watterson, worth at least a smile a day and the Montreal Canadiens, the best hockey team in the universe, for all the great memories.

Finally, I would like to express my gratitude to my parents, Jean-Marc and Cécile. I spent ten years in universities where I have learned a great deal, but things that really matter, I have learned from you.

Abstract

Vortex methods are a powerful tool for the simulation of incompressible flows at high Reynolds number. They rely on a discrete Lagrangian representation of the vorticity field to approximately satisfy the Kelvin & Helmholtz theorems which govern the dynamics of vorticity for inviscid flows.

A time splitting technique can be used to include viscous effects. The diffusion equation is considered separately after convecting the particles with an inviscid vortex method. In this thesis, the viscous effects are represented by the so-called deterministic method. The approach was extended to problems where a flux of vorticity is used to enforce the no-slip boundary condition. The ability of such a scheme to create the right amount of vorticity at the wall and to adequately redistribute it within the fluid is demonstrated by simulating the viscous flow induced by an oscillating cylinder.

In order to accurately compute the viscous transport of vorticity, gradients need to be well resolved. As the Reynolds number is increased, these gradients get steeper and more particles are required to achieve the requisite resolution. In practice, the computing cost associated with the convection step dictates the number of vortex particles and puts an upper bound on the Reynolds number that can be simulated with confidence.

That threshold can be increased by reducing the asymptotic time complexity of the convection step from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. The near-field of every vortex particle is identified. Within that region, the velocity is computed by considering the pairwise interaction of vortices. The speed-up is achieved by approximating the influence of the rest of the domain, the far-field. In that context, the interaction of two vortex particles is treated differently depending on their spatial relation. The resulting computer code does not lend itself to vectorization but has been successfully implemented on concurrent computers.

The combination of a fully viscous vortex method with a fast parallel algorithm is used to simulate the flow past an impulsively started cylinder. Experiments have shown that this flow is characterized by the presence of a secondary eddy within the main recirculating region. The numerical simulations successfully reproduced these secondary structures over a wide range of Reynolds number (Re=550 to 9500). It was observed that the secondary phenomenon can lead to a major flow reorganization by drastically altering the transport of vorticity. The separating boundary layer acts as a source of vorticity and, at Re=550, the resulting vortex sheet smoothly rolls up into the primary vortex. For Re=3000 and 9500, however, the secondary eddy interferes with that process and the flux of vorticity is redirected toward the cylinder where it accumulates into a new vortical structure.

The impulsive start is followed by a $1/\sqrt{T}$ singularity in the drag coefficients. The numerical simulations captured this behavior and the computed drag history for short times is in close agreement with the one predicted by a matched asymptotics analysis.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The equations of fluid mechanics are usually stated in terms of the velocity and pressure fields, the so-called primitive variables. There are, however, many advantages to recast these equations in terms of the vorticity and the streamfunction (Saffman 1981). This is especially true when the fluid density is a constant. The pressure then completely disappears from the problem formulation as pressure gradients cannot exert a torque on the fluid and affect its vorticity. Secondly, the knowledge of the vorticity, a scalar in two-dimensional flows, is sufficient to describe the flow field. In most applications, the vorticity is almost everywhere zero and one needs to compute its evolution only where it does not vanish.

Discrete vortex methods, first described by Rosenhead (1931), numerically exploit these attributes. More recent efforts are reviewed in Leonard (1980, 1985) and Sarpkaya (1989). These methods are based on the Kelvin & Helmholtz theorems which, in the inviscid limit, govern the dynamics of vorticity. The theorems state that vortex lines are advected at the local fluid velocity while a collection of these lines conserves its circulation. To solve the inviscid vorticity equation numerically, the vorticity field is first decomposed into discrete Lagrangian vortex elements. The velocity field is reconstructed and evaluated at the location of each Lagrangian particle. The Kelvin & Helmholtz theorems are then applied and the motion of the discrete elements is determined as if they were vortex lines. Actually moving the discrete representation of the vorticity field is a very natural way to deal with convection and, as a result, inviscid vortex methods are unconditionally stable. Unlike grid-based methods, no effort is spent computing the evolution of the irrotational portion of the flow. The behavior of the irrotational velocity field surrounding the vortical fluid, including the boundary condition at infinity, is built in the numerical scheme.

Different schemes have been proposed to solve the viscous vorticity equation. They all rely on the splitting of the time step into an inviscid and a viscous fractional step. In other words, convection and diffusion are dealt with sequentially. A conventional inviscid vortex method is first used to update the location of the vortices. Then, the discrete elements are not allowed to move while the diffusion equation is solved. The inclusion of viscous effects is challenging since the diffusion of vorticity must take place in a vorticity field described by Lagrangian particles.

In his review paper, Leonard (1980) observed that owing to its linearity, the diffusion equation can be identically satisfied by letting the vortex elements diffuse individually. The proper diffusive behavior is achieved by expanding the core of the regularized vortex blobs. While this method can represent the diffusion of the vorticity already present in the fluid, it does not provide a mechanism that would allow the transport of vorticity from the solid surfaces to the fluid. Moreover, diffusion produces fatter vortex blobs resulting in larger errors in the convection step; a fact that was later discussed by Greengard (1985). Nevertheless, this approach is appropriate for unbounded flows where the viscosity is such that the cores do not significantly expand during the course of the simulation.

In 1973, Chorin proposed to model viscous effects by imposing a random walk to each vortex element. One of the appealing features of this method is that it provides a framework that allows the transfer of vorticity from the walls to the fluid. In bounded flows, the no-slip condition is enforced by creating new vortices at the solid surfaces. These new vortices are themselves subjected to a random walk that takes them from the body to the fluid. The vorticity creation occurs within the boundary layers where the vorticity gradients are decisively steeper in the direction normal to the wall. Noting this fact, Chorin (1978b) subsequently modified the previous method and used Lagrangian vortex sheets instead of vortex blobs in the boundary layers. The vortex sheets are arbitrarily transformed into blobs when they leave the vicinity of the body. For a number of years, these methods have been the workhorses of the vortex simulation of bluff body flows.

There is no doubt that applying a random walk to a large number of particles can indeed mimic the diffusive effects of viscosity. It is another matter, however, to accurately represent these effects with limited computing resources. Milinazzo & Saffman (1977) have shown that a large number of vortex particles is required to adequately capture the diffusion of a single Gaussian vortical core. Despite

Chorin's objections (1978a), the fact remains that a reasonable number of particles can hardly handle the simplest diffusive processes. To imply that the random walk of the same limited number of vortex particles can account for the complex viscous interactions involved in a bluff body flows requires nothing short of an act of faith.

Those who could not act on that faith had no other choices but to limit themselves to effectively inviscid computations. That was until Raviart (1985, 1987) and his co-workers, Degond & Mas-Gallic (to appear) and Mas-Gallic (1985), noted that the Laplacian can be replaced by an integral operator. The discrete representation of the vorticity field is in turn substituted in that operator yielding an evolution equation for the circulation of every vortex particle. This group of investigators has shown that under certain conditions (including an infinite domain), this set of equations converges to the diffusion equation at a rate that depends on the choice of regularization function. The statistical treatment of viscous effects was so entrenched in some segments of the vortex methods community that this approach was immediately dubbed as "deterministic." Most methods are. Since it involves a rearrangement of the particles' weight, this approach will be referred to as the weight redistribution method throughout this document. Chap. **2** shows that the integral representation of the Laplacian can be derived by directly differentiating the regularized vorticity field. Besides demystifying the origin of the integral representation, this procedure can also be extended to problems where a vorticity flux is used to enforce the no-slip condition. The latter is essential for the simulation of bluff body flows.

Another "deterministic" treatment of viscous diffusion was proposed by Degond & Mustieles (1990). In this approach, the diffusive transport is approximated by displacing the particles along the vorticity gradient. This method is very robust and can be extended to problems with a high dimensionality. For the Navier-Stokes equations, the authors themselves recommend the use of a viscous operator based on the weight of the particles.

It is important to solve the diffusion equation with as few particles as possible since they have to be convected as well. The velocity induced by any given discrete vortex element is felt throughout the domain and, specifically, at all the other particles location. The most natural way to evaluate the velocity of a particle is to sum the contribution of all the others. The resulting $\mathcal{O}(N^2)$ asymptotic computational complexity has put a severe upper bound on the number of particles that can be handled by even the fastest computers*. In effect, the availability of computer resources or, more exactly, the lack of it, limits the size of the length scales that can be resolved and, consequently, the magnitude of the Reynolds number that can be simulated with confidence.

Faster computers can delay the inevitable but the appetite of the $\mathcal{O}(N^2)$ approach is such that doubling the computing speed only allows a marginal gain in the Reynolds number. Another, more promising, option is to introduce approximations that would speed-up the velocity evaluations.

The quest for speed began with Birdsall & Fuss' cloud-in-cell method (1969) which was first applied to vortex dynamics by Christiansen (1973a, b). In this particle-mesh approach, the particles do not interact directly. Instead, the vortices circulations are projected onto a fixed Eulerian grid and a fast Poisson solver is used to determine the velocity at the grid points. These velocities are then interpolated back to the particle locations. If the long-range interactions are computed properly, it is unclear how the interpolation and projection operators affect small scale motions.

This shortcoming was overcome by Eastwood & Hockney (1974, 1981) with a particle-particle/particle-mesh approach (P^3M). The mesh is used to compute the slow varying long-range interactions only. The near-field of every particle is treated exactly with particle to particle interactions. Along those lines, Anderson (1986) introduced the method of local corrections. An approximate velocity field is first computed on the Eulerian mesh by assuming a singular distribution of vorticity and using a classical cloud-in-cell approach. For each particle, the influence of neighboring vortices is then evaluated at the appropriate grid points and removed from the approximate velocity field before it is interpolated to the particles location. Pairwise interactions of regularized vortex blobs are then used to compute the influence of the near-field exactly. Anderson also took advantage of the harmonicity of the velocity field to construct a more accurate interpolation scheme. The resulting time complexity is $\mathcal{O}(M\log M + N)$ where M is the number of grid points in the Eulerian mesh. The mesh has to become finer as N is increased, so the method is really $\mathcal{O}(N\log N)$. The method of local corrections was successfully transported to concurrent computers by Baden (1987).

---

* Typically a few thousand.

The most recent fast velocity solvers are based on multi-range approximations. Again, the long-range interactions are approximated while the near-field is computed exactly. Instead of using a fast Poisson solver, these algorithms rely on interactions between groups of computational elements. For example, Spalart & Leonard (1981, 1983) covered the computational domain with square cells. The content of each cell, i.e. its vortices, is described by a multipole expansion. This approximate description is used when non-adjacent cells interact. Within the cell where the influence of a multipole expansion is sought, the velocity is expressed as a Taylor series centered at the center of that cell. The number of terms kept in both the Taylor and multipole expansions depends on the spatial relation of the pair of cells that is considered. Once again, the scheme switches back to pairwise interactions of vortices to compute the mutual influence of adjacent cells. The resulting time complexity is $\mathcal{O}(N^{\frac{4}{3}})$ but when vectorization is taken into account, the classical approach is still faster in most situations.

This method had all the ingredients that make $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ simulation possible, except the hierarchical data structure. This key element was introduced by Appel (1985). The size of the cells is adjusted to reflect the distance that separates them. The number of terms in the expansions is kept constant. Before the velocity evaluation per se, groups of different sizes are identified and organized into a binary tree data structure. The smallest ones consist of a few vortices only. The largest includes the whole computational domain. When computing the influence of a portion of the domain on another, one tries to use the largest groups first. The coarse description of the vorticity field that they provide might be sufficient if the groups in question are far apart. If these two groups are too close, smaller ones are sought. The binary tree data structure provides a quick way to access the relevant information. Ultimately, the near-field of every particle is treated with pairwise interactions of vortices.

Appel (op. cit.) used a low order approach keeping only two terms in the expansion of the inducting group and one term in the region where the velocity is sought (in effect, the induced velocity is assumed to be constant throughout the group). Greengard & Rokhlin (1987) generalized this approach to expansions of arbitrary order and put an upper bound on the error resulting from any given approximation. When considering the interaction of two groups, that error estimate is evaluated and compared to an accuracy criterion. If too large an error would result from computing the interaction at that level, the groups are subdivided into smaller ones and new error estimates are obtained. The process is repeated until the accuracy criterion is met or until the smallest groups are reached in which case, the algorithm switches back to particle to particle interactions.

Because the interaction of vortices is treated differently depending on their spatial relation, the vectorization of hierarchical methods is difficult. Concurrent computing is an attractive alternative and Chap. **3** describes the parallel implementation of a fast algorithm that combines Appel's data structure with Greengard & Rokhlin (op. cit.) high order expansions. The Barnes & Hut scheme (1986), which is basically another variation on Appel's paper, was also successfully implemented on parallel computers by Salmon (1990).

The combination of a fully viscous vortex method with the rapid evaluation of the velocity field provides a powerful tool for the simulation of incompressible flows past bluff bodies. Simulations of the symmetric development of the wake left behind an impulsively started circular cylinder are presented in Chap. **4**. This is a very compelling test case as a simple motion applied to a simple geometry results in complex fluid motions. The existence and behavior of these motions is well documented experimentally dating back to Prandtl (1904). Some of his flow visualizations are also available in Prandtl & Tietjens (1934). More recently, this problem was revisited by Bouard & Coutanceau (1980), Nagata, Funada, Kawai & Matsui (1985) and by Nagata, Nagase & Ito (1990). These experiments have revealed the existence of secondary eddies enclosed within the main recirculating region. The nature of the secondary phenomena is highly dependent on Reynolds number and provides a challenging test for numerical schemes. The numerous attempts to compute this flow, either theoretically or numerically, will be briefly described. For a broader discussion of unsteady fluid motion, the reader is referred to McCroskey (1977).

Historically, the flow past an impulsively started cylinder was first transformed into a more tractable problem by applying the boundary layer approximations. The impulsive start is followed by the diffusion of the vortex sheet generated by the irrotational flow. Blasius (1908) chose a series solution in powers of time and showed that, to first order, that diffusion proceeds locally like a Rayleigh solution indicating that the diffusion of vorticity normal to the wall dominates the convection term and the diffusion along the wall. He also determined a second order correction to this solution. Using a similar technique, Goldstein & Rosenhead (1936) carried the analysis to one more order of accuracy. Alternatively, the boundary layer equations can be integrated numerically, as first described by Stewartson and his co-workers (1971). Methods based on the integral representation of the unsteady boundary layer equations were also applied to the impulsively started cylinder (Schuh 1953). These studies, and most of the ones published since, have been concerned with the emergence of a recirculating region. Since the outer flow is not coupled with the growth of the boundary layer, the predicted separation time does not depend on the Reynolds number and that time can be thought of as an asymptotic limit, valid at high Reynolds numbers.

This constraint can be relaxed by solving the full Navier-Stokes equations. From the theoretical perspective, inner and outer expansions matched to the third order were obtained by Wang (1967) for the primitive variables and by Bar-Lev & Yang (1975) in the vorticity-streamfunction formulation. These solutions are valid for short times where the convection of vorticity is dominated by diffusion normal to the wall. To go further in time, the Navier-Stokes equations have to be integrated numerically.

Twenty-five years after Thom's (1933) simulation of the steady flow past a cylinder, Payne (1958) used finite differences to solve the unsteady equation for the impulsive motion of the cylinder at $Re = 40$ and 100. Ingham (1968) performed simulations at the same Reynolds numbers and showed a total drag coefficient that reaches a maximum at $T \simeq 4$. Son & Hanratty (1969) pushed the Reynolds number up to 500 but did not observe the formation of secondary eddies. However, these structures were present in Patel's simulation (1976), obtained with a truncated sine series in θ and a discrete representation of the associated shape functions in the radial direction. More recently, Ta Phuoc Loc (1980) pushed the Reynolds number up to 1000 and later collaborated with Bouard (1985) to produce simulations that agree remarkably well with the Bouard & Coutanceau's experiments (op. cit.) at $Re = 3000$ and 9500. In both cases, fourth order compact finite differences are used to solve the vorticity equation. The full numerical procedure is described in Daube & Ta Phuoc Loc (1978).

The flow past a cylinder is a favorite of the vortex methods community. The shape is appealing because image vortices can be used to enforce the inviscid boundary condition. At the same time, it presents the investigators with a fundamental difficulty: where should the vorticity be allowed to leave the body? This problem, common to all bodies devoid of sharp edges, was recognized very early by Gerrard (1967). Different approaches have since been used to deal with the unknown separation points and have been extensively reviewed in Sarpkaya (1989). For effectively inviscid simulations, the treatment of the primary separation falls into either of the following categories:

- fixed separation point, most often suggested by experimental data,
- arbitrary criterion generally derived from the knowledge of the velocity field,
- separation point determined from a boundary layer calculation (see Spalart, Leonard & Baganoff 1983).

At moderate Reynolds number, vortex methods can be used to compute both the wake and the boundary layer (see Cheer 1983, 1989). It should be noted that the boundary layer approximations are not used in the simulations presented in Chap. 4. As in the work of Stansby & Smith (1989), the full vorticity equation is solved throughout the domain removing the need for a matching procedure.

To recapitulate, Chap. **2** presents an extension of the "deterministic" method to bounded flow; Chap. **3** describes a fast velocity solver and its implementation on concurrent computers and Chap. **4** applies the resulting numerical scheme to the computation of a viscous flow past an impulsively started cylinder. The numerical simulations are compared to published experimental observations, mainly those of Bouard & Coutanceau which seem to have become the yardstick upon which the ability of numerical schemes to simulate bluff body flows is measured. At $Re = 9500$, a marginal numerical resolution makes the simulation suspect, especially at early times but for the two lower Reynolds numbers, $Re = 550$ and 3000, the agreement between the computed streamlines and the flow visualizations is found to be very good. Furthermore, the asymptotic behavior of the drag coefficient near $T = 0$ matches the one theoretically predicted by Bar-Lev & Yang (1975). Within the aforementioned range of Reynolds numbers, viscous vortex methods have become a viable alternative to finite differences.

CHAPTER 2

A Viscous Vortex Method

And God created the two-dimensional incompressible Navier-Stokes equations for a Newtonian fluid,

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

$$\nabla \cdot \mathbf{u} = 0 \ , \tag{2.2}$$

within the domain Ω and with the boundary condition $\mathbf{u} = \mathbf{u}_{\text{wall}}$ on the solid surfaces identified as Γ. Man thought they were good but soon, the advantages of recasting them in terms of the vorticity,

$$\omega = \nabla \times \mathbf{u} \ , \tag{2.3}$$

became apparent*. An equation for the time evolution of this new quantity can be found by taking the curl of Eq. (2.1) subject to the divergence free constraint of Eq. (2.2),

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \nabla^2 \omega \ . \tag{2.4}$$

To solve this equation, it is proposed to use a time-splitting scheme. During the first fractional step, viscous effects are ignored and the Euler equation is solved as the vorticity field is allowed to convect. That displaced field is then frozen and the diffusion of vorticity takes place in this motionless environment. Beale & Majda (1981) have shown that the split scheme converges to the Navier-Stokes equations at the rate $C\nu\Delta t$, where C is a problem-dependent proportionality constant. Symbolically, the scheme can be written as

---

* the reader should notice the equal treatment given to creation and evolution.

$$\text{first fractional step : } \left\{ \begin{array}{rl} \dfrac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = 0 & \text{in } \Omega \\ \mathbf{u} \cdot \mathbf{n} = 0 & \text{on } \Gamma \end{array} \right\} \tag{2.5}$$

$$\text{second fractional step : } \left\{ \begin{array}{ll} \dfrac{\partial \omega}{\partial t} = \nu \nabla^2 \omega & \text{in } \Omega \\ \nu \dfrac{\partial \omega}{\partial n} & \text{known on } \Gamma \end{array} \right\} . \tag{2.6}$$

The first fractional step is solved by a conventional inviscid vortex method, as described extensively in Christiansen (1973), Leonard (1980) and Sarpkaya (1989). The region occupied by rotational fluid is discretized into Lagrangian vortex particles, i.e.

$$\omega(\mathbf{x}, t) = \sum_i^N \alpha_i(t) R_\sigma(\mathbf{x} - \mathbf{x}_i(t)) \ . \tag{2.7}$$

Each particle is responsible for a small area of fluid; it carries a circulation, α_i, which is the integral of vorticity over that area. The distribution of vorticity within an isolated vortex particle is determined by R_σ, the regularization or cut-off function. To conserve circulation, this function must satisfy

$$\iint_\Omega R_\sigma(\mathbf{x})\, d\mathbf{x} = 1 \ , \tag{2.8}$$

or

$$2\pi \int_0^\infty R_\sigma(r)\, r\, dr = 1 \ , \tag{2.9}$$

when it is radially symmetric. The subscript "σ" indicates that the argument of R has been normalized with the core size of the vortex blob, σ, and as a consequence of Eq. (2.8), it can be shown that R_σ may be written as

$$R_\sigma(\mathbf{x}) = \frac{1}{\sigma^2} R\left(\frac{\mathbf{x}}{\sigma}\right) \ . \tag{2.10}$$

In the simplest case, R_σ is a Dirac delta function. The vorticity field is then singular and the discrete vortex elements are referred to as point vortices. This thesis will be concerned with smooth vorticity fields and the cut-off function will either be a Gaussian or an algebraic approximation of the Gaussian.

In the absence of viscosity, Kelvin's and Hemholtz's theorems apply. They state that vortex tubes conserve their circulation while moving at the local fluid velocity. In two dimensions, the vortex blobs can be thought of as the cross-section of infinite vortex tubes perpendicular to the plane of computation, and the theorems are used to update the particle locations. To close the loop, velocity has to be evaluated at the center of each vortex blob. As discussed in Chap. 3, the velocity field can be reconstructed from a known vorticity distribution by solving

$$\nabla^2 \psi = -\omega \quad , \tag{2.11}$$

where ψ is the stream function with

$$\mathbf{u} = \nabla \times (\psi \hat{\mathbf{e}}_z) \quad . \tag{2.12}$$

The inviscid boundary condition is enforced with a panel type method or when the geometry is simple (i.e., half-plane, circle, etc.), image vortices can also be used.

Viscous effects are modeled by an exchange of circulation between the vortex blobs. This exchange takes place in the second fractional step and will be discussed in greater detail in the next section. In essence, the weights of the particles are redistributed in a way that is consistent with the diffusive action of viscosity. Simultaneously, the vorticity emanating from the solid surfaces is appropriately assigned to nearby fluid particles. The flux of vorticity**, $\nu \frac{\partial \omega}{\partial n}$, is actually derived from the no-slip condition using a procedure similar to the one proposed by Chorin (1973). At the end of the first fractional step, a slip velocity is observed at the wall. Then, the vorticity flux is chosen in such a way that the no-slip boundary condition will be enforced at the end of the second fractional time step, i.e.

$$\nu \frac{\partial \omega}{\partial n} = \frac{u_{\text{slip}}}{\Delta t} \quad . \tag{2.13}$$

** **n is the vector normal to the solid surface and is positive when pointing into the fluid.**

2.1 A particle representation of diffusion

One of the most significant advantages of the vortex method over its "competitors" is that it is essentially grid free. The user doesn't have to worry about generating body-fitted grids, and the Lagrangian nature of the method makes it unconditionally stable to convection. However, what makes the vortex method attractive for convection dominated flows, also makes the inclusion of viscous effects very difficult. Even if the Lagrangian grid is well organised at $t = 0$, the shearing and stretching due to convection will rapidly distort that grid and make it unfit for the classical approaches to diffusion[†]. This section will present a numerical scheme capable of solving the diffusion equation in the absence of a structured grid. It will rely exclusively on the local knowledge of the vorticity field carried by the Lagrangian particles. It will be assumed that there is a sufficient number of particles to densely cover the region where the vorticity is non zero.

Since the scheme involves a readjustment of the particle strengths or weights, it will be referred to as the weight redistribution method (WRM). The method involves a regularized vorticity field, $\omega_R(\mathbf{x}, t)$, which is derived from the known vorticity field, $\omega(\mathbf{x}, t)$, using the following convolution:

$$\omega_R(\mathbf{x}, t) = \iint_\Omega \omega(\mathbf{y}, t) R_\sigma\left(|\mathbf{x} - \mathbf{y}|\right) \, d\mathbf{y} \ . \tag{2.1.1}$$

This expression is basically a smoothing operator that removes all wavelengths which are small compared to σ; the larger scales are left intact. The accuracy of the approximation depends on the moment properties of the regularization kernel $R_\sigma(|\mathbf{x} - \mathbf{y}|)$. A Gaussian kernel, for example, introduces an error of $\mathcal{O}(\sigma^2)$. Eq. (2.1.1) can also be used to transform a point vortex into a blob.

To find a time evolution equation for the regularized vorticity field, one can follow Monaghan's suggestion, (1982, 1983), and write the diffusion equation in the $\mathbf{y}$-space,

$$\frac{\partial \omega(\mathbf{y}, t)}{\partial t} = \nu \nabla_{\mathbf{y}}^2 \omega(\mathbf{y}, t) \ . \tag{2.1.2}$$

[†] finite differences, finite elements, etc. .

The $\mathbf{y}$ subscript in $\nabla_{\mathbf{y}}$ expresses derivatives taken with respect to the dummy variable $\mathbf{y}$. The smoothing operator, $R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)$, is then applied to Eq. (2.1.2) and its left hand side becomes the time derivative of $\omega_R(\mathbf{x},t)$,

$$\begin{aligned}\iint_\Omega \frac{\partial\omega(\mathbf{y},t)}{\partial t} R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} &= \frac{\partial}{\partial t}\iint_\Omega \omega(\mathbf{y},t)R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} \\ &= \frac{\partial\omega_R(\mathbf{x},t)}{\partial t}\ .\end{aligned} \tag{2.1.3}$$

Applying the same procedure to the right hand side and integrating by parts yields

$$\begin{aligned}\iint_\Omega \nu\nabla^2_{\mathbf{y}}\omega(\mathbf{y},t)R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} &= \nu\iint_\Omega \nabla_{\mathbf{y}}\cdot\left(R_\sigma(|\mathbf{x}-\mathbf{y}|)\nabla_{\mathbf{y}}\omega(\mathbf{y},t)\right)\,d\mathbf{y} \\ &\quad -\nu\iint_\Omega \nabla_{\mathbf{y}}\omega(\mathbf{y},t)\cdot\nabla_{\mathbf{y}}R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)d\mathbf{y} \\ &= \nu\int_\Gamma R_\sigma(|\mathbf{x}-\mathbf{y}_\Gamma|)\frac{\partial\omega(\mathbf{y}_\Gamma,t)}{\partial n}\,d\Gamma \\ &\quad -\nu\iint_\Omega \nabla_{\mathbf{y}}\omega(\mathbf{y},t)\cdot\nabla_{\mathbf{y}}R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y}\ .\end{aligned} \tag{2.1.4}$$

Finally,

$$\begin{aligned}\frac{\partial\omega_R(\mathbf{x},t)}{\partial t} &= \nu\int_\Gamma R_\sigma\left(|\mathbf{x}-\mathbf{y}_\Gamma|\right)\frac{\partial\omega(\mathbf{y}_\Gamma,t)}{\partial n}\,d\Gamma \\ &\quad -\nu\iint_\Omega \nabla_{\mathbf{y}}\omega(\mathbf{y},t)\cdot\nabla_{\mathbf{y}}R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y}\ .\end{aligned} \tag{2.1.5}$$

The term $\nu\frac{\partial\omega}{\partial n}$ is the flux of vorticity emanating from the solid surfaces. Such a flux can be created by accelerating the surfaces or by a pressure gradient along them. The first term specifies how to distribute that flux within the fluid in a way that is consistent with the approximation introduced in Eq. (2.1.1). The second term deals with the diffusion of vorticity within the fluid interior. It can be further simplified by noting that the kernel has radial symmetry which implies that

$$\nabla_{\mathbf{y}} R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right) = \frac{1}{\sigma^2}\frac{(\mathbf{x}-\mathbf{y})}{\rho}\frac{dR_\sigma(\rho)}{d\rho} \quad , \tag{2.1.6}$$

where

$$\rho = \frac{|\mathbf{x}-\mathbf{y}|}{\sigma} \quad . \tag{2.1.7}$$

Defining

$$\eta_\sigma\left(|\mathbf{x}-\mathbf{y}|\right) = -\frac{1}{\rho}\frac{dR_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)}{d\rho} \tag{2.1.8}$$

and shifting the origin of $\mathbf{y}$ to $\mathbf{x}$, the second term becomes

$$-\nu\iint_\Omega \nabla_{\mathbf{y}}\omega(\mathbf{y},t)\cdot\nabla_{\mathbf{y}}R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} = \frac{\nu}{\sigma^2}\iint_\Omega \mathbf{y}\cdot\nabla_{\mathbf{y}}\omega(\mathbf{x}+\mathbf{y},t)\eta_\sigma(|\mathbf{y}|)\,d\mathbf{y} \quad . \tag{2.1.9}$$

Since the kernel η_σ rapidly decreases away from the origin, only small values of $|\mathbf{y}|$ will contribute to the integral and $\nabla_{\mathbf{y}}\omega(\mathbf{y}+\mathbf{x},t)$ can be replaced by a Taylor series centered a $\mathbf{x}$. A first order backward difference,

$$\mathbf{y}\cdot\nabla_{\mathbf{y}}\omega(\mathbf{x}+\mathbf{y},t) \simeq \omega(\mathbf{y}) - \omega(\mathbf{x}) \quad , \tag{2.1.10}$$

could be used but high order of spatial accuracy cannot be obtained with such a crude approximation. When a Gaussian or an equivalent smoothing function is used, a second order backward difference is more appropriate;

$$\begin{aligned} \mathbf{y}\cdot\nabla_{\mathbf{y}}\omega(\mathbf{x}+\mathbf{y},t) &\simeq \frac{3}{2}\omega(\mathbf{y}) - 2\omega(\mathbf{x}) + \frac{1}{2}\omega(-\mathbf{y}) \\ &\simeq \frac{3}{2}\big(\omega(\mathbf{y})-\omega(\mathbf{x})\big) + \frac{1}{2}\big(\omega(-\mathbf{y})-\omega(\mathbf{x})\big) \quad . \end{aligned} \tag{2.1.11}$$

Substituting expression (2.1.11) into Eq. (2.1.9) and using the symmetry of the heat kernel, η_σ, it is found that

$$-\nu \iint_\Omega \nabla_{\mathbf{y}}\omega(\mathbf{y},t)\cdot\nabla_{\mathbf{y}}R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} \simeq \frac{2\nu}{\sigma^2}\iint_\Omega \left[\omega(\mathbf{y},t)-\omega(\mathbf{x},t)\right]\eta_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\,d\mathbf{y} \tag{2.1.12}$$

which is the expression used by Mas-Gallic (1985), Raviart (1985, 1987), Degond & Mas-Gallic (to appear) and Winckelmans (1988, 1989) to represent viscous diffusion in an infinite domain. When boundaries are present, it is found that

$$\begin{aligned}\frac{\partial\omega_R(\mathbf{x},t)}{\partial t} \simeq \nu \int_\Gamma R_\sigma\left(|\mathbf{x}-\mathbf{y}_\Gamma|\right)\frac{\partial\omega(\mathbf{y}_\Gamma,t)}{\partial n}\,d\Gamma \\ +\frac{2\nu}{\sigma^2}\iint_\Omega \eta_\sigma\left(|\mathbf{x}-\mathbf{y}|\right)\left[\omega(\mathbf{y})-\omega(\mathbf{x})\right]d\mathbf{y}\ .\end{aligned} \tag{2.1.13}$$

The same procedure can now be applied to a discrete version of the vorticity field,

$$\omega^h(\mathbf{x},t) = \sum_i^N \alpha_i(t)\,\delta\big(\mathbf{x}-\mathbf{x}_i(t)\big)\ , \tag{2.1.14}$$

where

$$\alpha_i(t) = h_i^2\,\omega(\mathbf{x}_i,t)\ . \tag{2.1.15}$$

The superscript h refers to the typical distance between two adjacent particles and h_i^2 is the fluid area associated with particle "i." It should be noted that since the fluid is incompressible, the area associated with a given particle is not a function of time. When this singular vorticity field is regularized by the operator (2.1.1), Eq. (2.1.13) can be applied at the location of particle "i" and it is found that

$$\begin{aligned}\frac{\partial\omega_R^h(\mathbf{x}_i,t)}{\partial t} \simeq h_i^2\nu \int_\Gamma R_\sigma\left(|\mathbf{x}_i-\mathbf{y}_\Gamma|\right)\frac{\partial\omega(\mathbf{y}_\Gamma,t)}{\partial n}\,d\Gamma \\ +\frac{2\nu}{\sigma^2}\sum_j^N \eta_\sigma\left(|\mathbf{x}_i-\mathbf{x}_j|\right)\left[\alpha_j(t)h_i^2-\alpha_i(t)h_j^2\right]\ .\end{aligned} \tag{2.1.16}$$

Assuming that the time derivative of the regularized field can be applied to the discrete field, the time derivative of $\alpha_i(t)$ is found to be

$$\begin{aligned}\frac{d\alpha_i(t)}{dt} \simeq\ & h_i^2\nu \int_\Gamma R_\sigma\left(|\mathbf{x}_i - \mathbf{y}_\Gamma|\right) \frac{\partial\omega(\mathbf{y}_\Gamma, t)}{\partial n}\, d\Gamma \\ & + \frac{2\nu}{\sigma^2} \sum_j^N \eta_\sigma\left(|\mathbf{x}_i - \mathbf{x}_j|\right) \left[\alpha_j(t)h_i^2 - \alpha_i(t)h_j^2\right] \quad .\end{aligned} \tag{2.1.17}$$

For simplicity, it will be assumed that all particles have the same area, h^2, so that Eq. (2.1.17) reduces to

$$\begin{aligned}\frac{d\alpha_i(t)}{dt} =\ & h^2\nu \int_\Gamma R_\sigma\left(|\mathbf{x}_i - \mathbf{y}_\Gamma|\right) \frac{\partial\omega(\mathbf{y}_\Gamma, t)}{\partial n}\, d\Gamma \\ & + \frac{2\nu h^2}{\sigma^2} \sum_j^N \eta_\sigma\left(|\mathbf{x}_i - \mathbf{x}_j|\right) \left[\alpha_j(t) - \alpha_i(t)\right] \quad .\end{aligned} \tag{2.1.18}$$

It would seem that such a scheme would have an $\mathcal{O}(N^2)$ time complexity since the time derivative of a particle strength depends on all the other particles in the domain. But the kernel $\eta_\sigma\left(|\mathbf{x}_i - \mathbf{x}_j|\right)$ vanishes for large arguments and the viscous influence of a particle is limited to its immediate neighborhood. A particle needs only to interact with a few neighbors and the time complexity is really $\mathcal{O}(N)$. In practice, a numerical quadrature is needed to integrate the boundary term. These additional approximations will be discussed in App. A.

2.2 Numerical Considerations

Any numerical procedure introduces parameters whose values have a significant impact on the quality of the numerical solution. The solution usually improves as the resolution, both in space and time, is increased (convergence). Most methods also involve other numerical artifacts that may or may not have a connection with the physics that is modeled. A typical vortex method has many of those. How many new vortices should be created every time step? What strength should they have? How far from the wall should they appear? When is it acceptable to merge nearby vortices? The answers given to these questions can significantly alter the final picture and usually, only "experience" will provide the right chemistry for a given problem. The parameter space is so vast (see Tiemroth 1985), that no one dares to explore it systematically.

When implementing the method described in the previous section, one has to face these kinds of choices. In addition to the spatial spacing, h, which should be as small as possible, one has to select a regularization function, $R_\sigma(|\mathbf{x}-\mathbf{y}|)$, the blob overlap, $\frac{\sigma}{h}$, and a time step. The next sections will provide some insights on how to select these parameters in a systematic way.

2.2.1 Regularization function and overlap

The radially symmetric regularization function, R_σ, is the key element of a viscous vortex method. Even in the inviscid case, it determines the convergence rate of the method. Leonard (1980) has shown that the numerical approximation of the convection term can be improved by using regularization functions where both signs of vorticity are present. Spectral-like convergence can even be obtained with a first order Bessel function.

However, Raviart (1987) pointed out that the heat kernel, $\eta_\sigma(\rho)$, associated with these high order regularization functions leads to unacceptable errors in the diffusive process. To yield an accurate solution, the condition $\eta_\sigma(\rho) \geq 0$ must be satisfied for all ρ. In addition, he also demonstrated that convergence is achieved only when $R_\sigma(\rho)$ and its associated $\eta_\sigma(\rho)$ are of even order. These restrictions leave second order kernels as the only option.

The Gaussian smoothing,

$$R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{|\mathbf{x}-\mathbf{y}|^2}{\sigma^2}} \quad , \tag{2.2.1}$$

meets all these requirements and is a natural choice since it is the kernel of the heat equation. Its associated $\eta_\sigma(\rho)$ is also a Gaussian and with the proper normalization, it is found that $R_\sigma(\rho) = \eta_\sigma(\rho)$ which might be of some computational benefit.

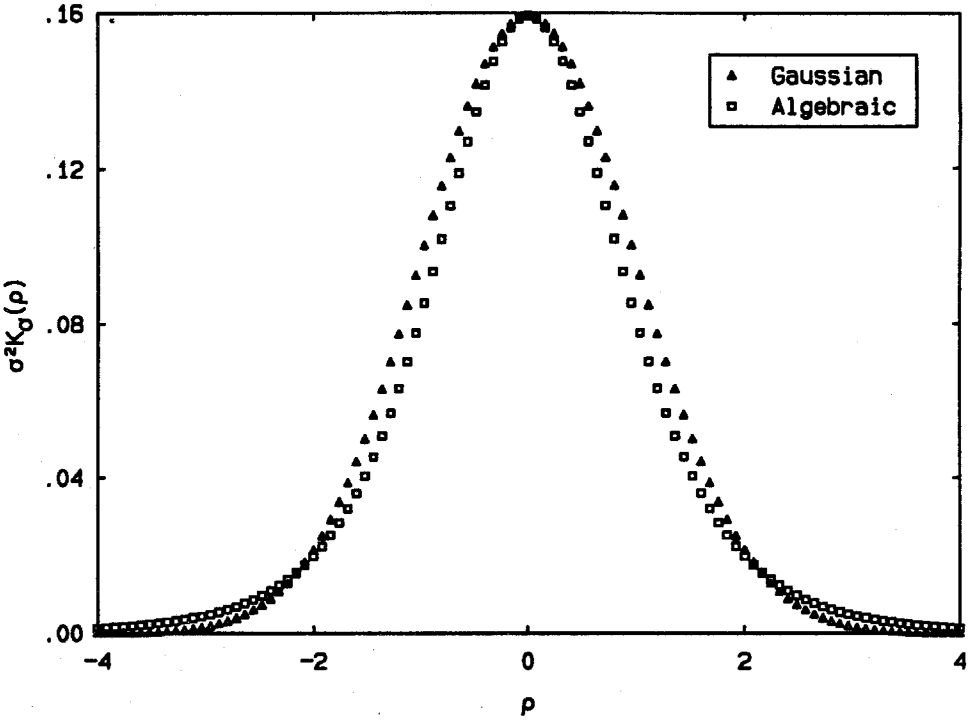


FIGURE 2.2.1 Comparison of Gaussian and algebraic regularization function.

A second order algebraic smoothing is also a valid choice and can be defined as

$$R_\sigma\left(|\mathbf{x}-\mathbf{y}|\right) = \frac{1}{2\pi\sigma^2} \frac{1}{\left(1+\frac{|\mathbf{x}-\mathbf{y}|^2}{4\sigma^2}\right)^3} \quad , \tag{2.2.2}$$

The factor 4 was introduced in the denominator to ensure that the condition

$$\int_0^\infty \eta_\sigma(\rho)\, d\rho = \int_0^\infty R_\sigma(\rho)\, d\rho \tag{2.2.3}$$

is respected as it is for the Gaussian. As shown if Fig. (2.2.1), the algebraic regularization function is a close approximation of the Gaussian although not quite as compact. As a result of these longer tails, smoothing and viscous effects are felt at greater distance from the particle. An algebraic function will have to be evaluated more often than its Gaussian counterpart. When look-up tables are used, the required time for an evaluation is the same for both functions and it is then faster to use a Gaussian regularization. The only reason to choose the algebraic smoothing is that it is easier to evaluate the integral of the boundary term in Eq. (2.1.18) (see App. A).

Raviart (1987) and Hald (1979) have shown that the overlapping of the blobs is necessary to obtain convergence to the Navier-Stokes equations as N, the number of particles, becomes large. It is another question, however, to determine how much overlap is needed to provide an accurate solution when a limited number of particles is used. To do so, the weight redistribution method was applied to the one-dimensional problem described in Fig. (2.2.2).

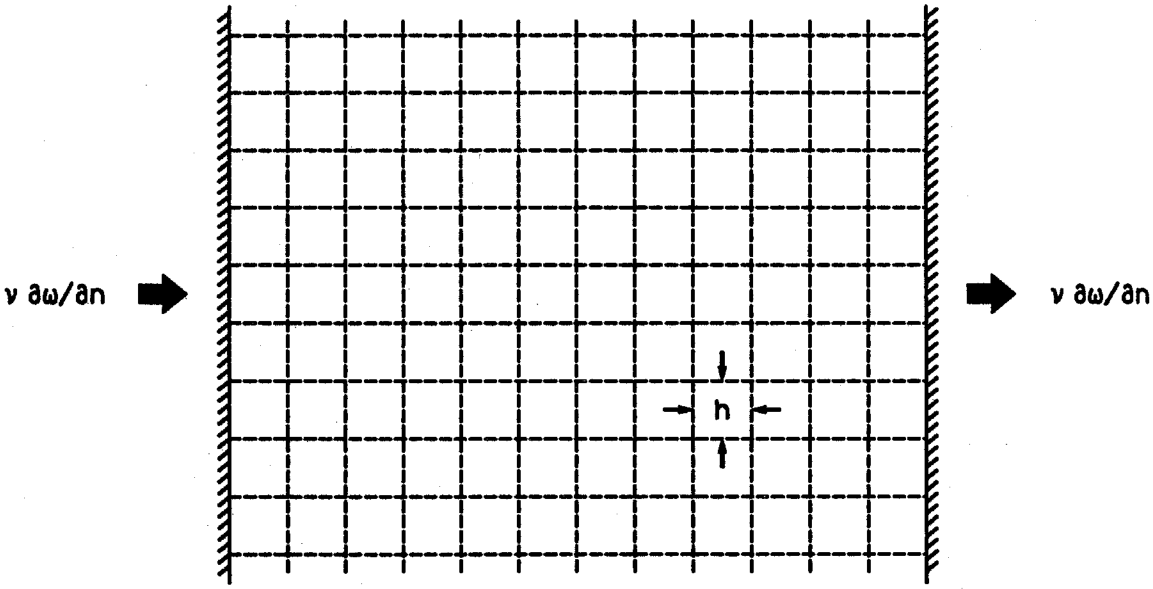


FIGURE 2.2.2 Overlap test case.

A constant vorticity flux enters the fluid from the left wall and leaves through the right one. Physically, this situation could be created by steadily accelerating the walls in opposite directions. A steady state will eventually be reached when a linear distribution of vorticity is established in the fluid corresponding to a uniform flux of vorticity, F_ω. Assuming that such a linear profile already exists, a numerical solution is sought by placing a regularized particle at the center of each $h \times h$ cell of Fig. (2.2.2). Modifying Eq. (2.1.18),the numerical flux crossing any vertical plane (far enough from the walls) can be written as

$$\hat{F}_\omega = \left(\frac{h}{\sigma}\right)^4 \sum_{i=1}^{M} \sum_{j=i}^{M} \sum_{k=-M}^{M} 2\nu \frac{\partial \omega}{\partial n} j \eta_\sigma\left(h^2(j^2+k^2)\right) , \qquad (2.2.4)$$

where M is large enough to include all the interactions for which η_σ is significant.

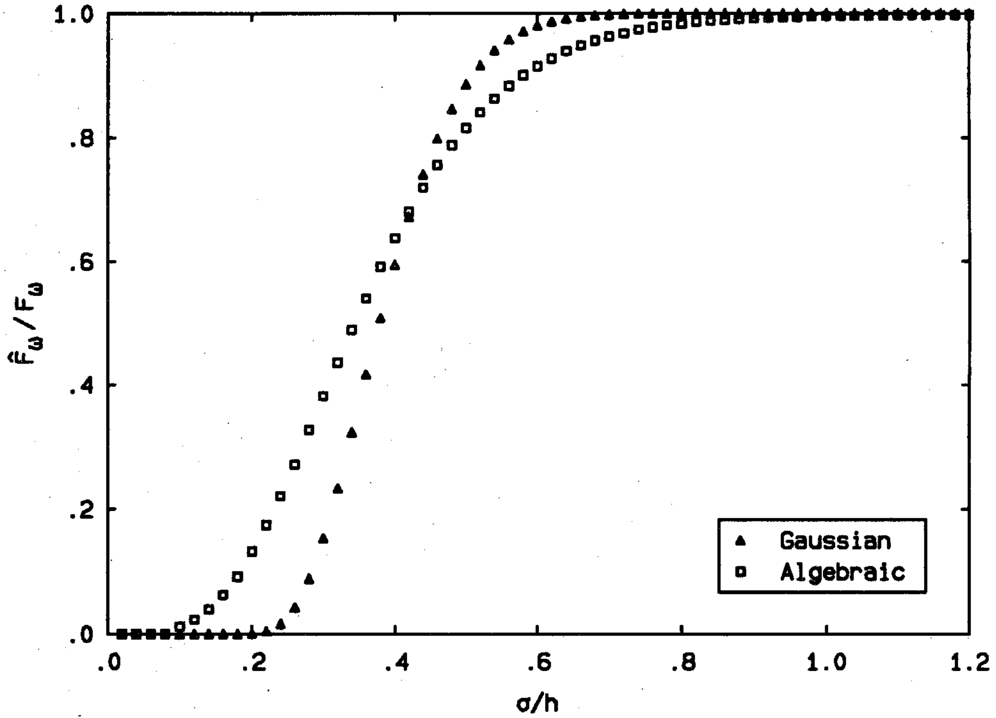


FIGURE 2.2.3 Effect of overlap on the numerical vorticity flux.

On Fig. (2.2.3), the numerical flux is plotted versus the overlap, $\frac{\sigma}{h}$. As expected, there can be no flux without overlap. Furthermore, it can be seen that vorticity is not properly exchanged (the flux is underestimated by more than 1%) when the overlap is less than $\simeq$ 0.6 for the Gaussian smoothing and $\simeq$ 0.8 for its algebraic counterpart. These minimum values have to be maintained at all times in any simulation that attempts to capture the viscous transport of vorticity. The gaussian kernel is slightly more forgiving in that respect which is another reason to select it over its algebraic cousin.

Finally, one should keep in mind that, although necessary, the overlap should be kept to a minimum since the spatial resolution is $\mathcal{O}(\sigma)$ while h dictates the computing cost. A delicate balance will have to be maintained between these conflicting requirements.

2.2.2 Time step

In this section, the effect of varying the time step on the accuracy of the WRM will be examined. For this purpose, the following two dimensional test case was selected. At $t = 0$, a Gaussian vorticity distribution with a standard deviation of 0.5,

$$\omega(x, y, \nu t=0) = \frac{2}{\pi} e^{-2(x^2+y^2)} \ , \tag{2.2.5}$$

is discretized by particles arranged on a 101 × 101 square grid. The grid covers a computational domain of approximately $[-4, 4] \times [-4, 4]$ and the blobs significantly overlap as $\frac{\sigma}{h} = 1$. The domain is considered to be infinite since the vorticity is essentially zero at its edges throughout the computation. Each particle is assigned an initial strength of

$$\alpha(x_i, y_i, \nu t=0) = h^2 \omega(x_i, y_i, \nu t=0) \ , \tag{2.2.6}$$

where h^2 is the area of one grid cell.

This initial vorticity distribution is then allowed to diffuse until it reaches a standard deviation of 1.0;

$$\omega(x, y, \nu t=0.375) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} \ . \tag{2.2.7}$$

The problem is purely diffusive and there are no walls where vorticity could be created or canceled. A numerical solution, $\hat{\omega}$, is obtained using the WRM and a forward Euler time integration scheme. The error between the exact solution and $\hat{\omega}$ is sampled at the blob locations only and is defined as

$$\Upsilon = \sqrt{\frac{\sum_i^N h^2\left(\hat{\omega}(x_i,y_i) - \omega_{\text{exact}}(x_i,y_i)\right)^2}{\sum_i^N h^2 \omega_{\text{exact}}^2(x_i,y_i)}} \; . \tag{2.2.8}$$

The behavior of Υ as a function of the normalized time step is shown in Fig. (2.2.4). A linear behavior was expected since forward Euler is accurate to $\mathcal{O}(\Delta t)$. However, the most accurate results are not obtained with the smallest time step. This remarkable result implies that the error due to the discrete time integration somewhat cancels the spatial error associated with the WRM. With Gaussian smoothing, this cancellation seems almost perfect when $\Delta t = \frac{\sigma^2}{2\nu}$.

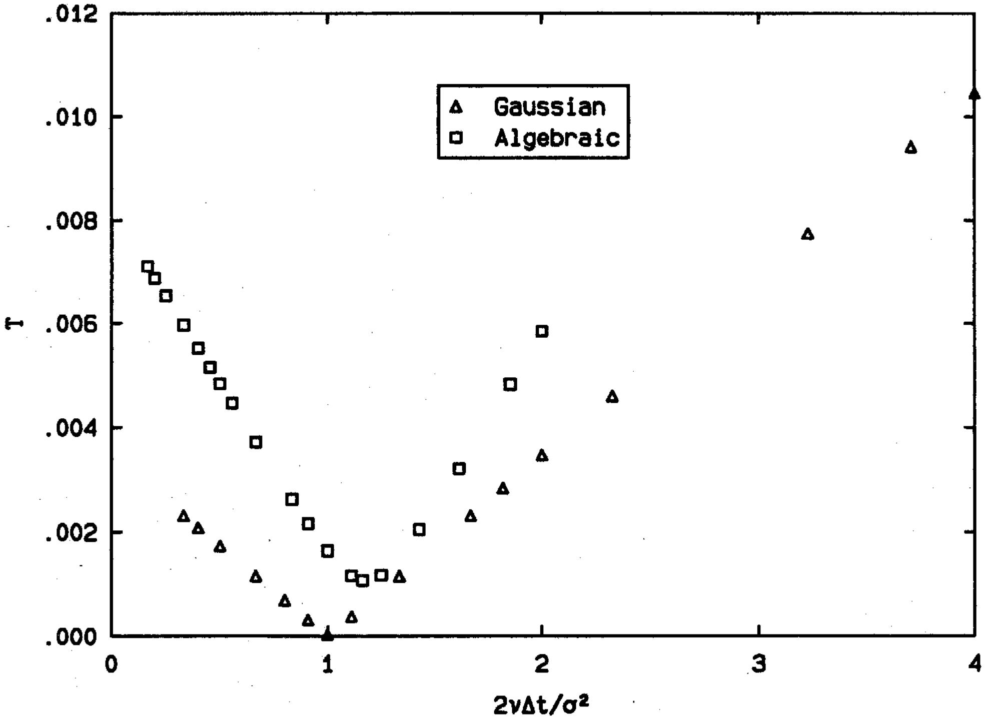


FIGURE 2.2.4 Accuracy of the weight redistribution method.

This phenomenon can be understood by applying the continuous form of the WRM to the two dimensional diffusion equation;

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u = \frac{2\nu}{\sigma^2} \iint_\Omega \eta_\sigma \left(|\mathbf{x} - \mathbf{y}| \right) \left[u(\mathbf{y}) - u(\mathbf{x}) \right] d\mathbf{y} \ . \tag{2.2.9}$$

Consider the behavior of a single Fourier mode,

$$u(\mathbf{x}, t) = a(t)\, e^{i\mathbf{k}\cdot\mathbf{x}} \ , \tag{2.2.10}$$

where

$$\mathbf{k} = k e^{i\phi} \ . \tag{2.2.11}$$

It is found that any mode decays exponentially in time according to

$$u(\mathbf{x}, t) = a_k e^{i\mathbf{k}\cdot\mathbf{x}} e^{-\nu k^2 t} \ . \tag{2.2.12}$$

Discretizing in time, the amplitude of mode "k" behaves like

$$a_k^{n+1} = a_k^n e^{-\nu k^2 \Delta t} \ . \tag{2.2.13}$$

Applying Eq. (2.2.9) with a Gaussian regularization function,

$$\eta_\sigma \left(|\mathbf{x} - \mathbf{y}| \right) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{|\mathbf{x}-\mathbf{y}|^2}{\sigma^2}} \ , \tag{2.2.14}$$

to the same Fourier mode, it can be shown that

$$\begin{aligned}
e^{i\mathbf{k}\cdot\mathbf{x}} \frac{da_k(t)}{dt} &= \frac{2\nu a_k(t)}{\sigma^2} \iint_\Omega \frac{1}{\sigma^2 2\pi} e^{\frac{-|\mathbf{x}-\mathbf{y}|^2}{2\sigma^2}} \left[e^{i\mathbf{k}\cdot\mathbf{y}} - e^{i\mathbf{k}\cdot\mathbf{x}} \right] d\mathbf{y} \\
&= \frac{2\nu a_k(t)}{\sigma^2} \left[-e^{i\mathbf{k}\cdot\mathbf{x}} + \frac{e^{i\mathbf{k}\cdot\mathbf{x}}}{2\pi\sigma^2} \iint_\Omega e^{-\frac{1}{2}\frac{|\mathbf{z}|^2}{\sigma^2}} e^{i\mathbf{k}\cdot\mathbf{z}}\, d\mathbf{z} \right] \\
&= \frac{2\nu a_k(t)}{\sigma^2} e^{i\mathbf{k}\cdot\mathbf{x}} \left[-1 + \int_0^\infty \frac{1}{2\pi\sigma^2} r e^{-\frac{1}{2}\frac{r^2}{\sigma^2}} \int_0^{2\pi} e^{ikr\cos(\theta-\phi)}\, d\theta\, dr \right] \\
&= \frac{2\nu a_k(t)}{\sigma^2} e^{i\mathbf{k}\cdot\mathbf{x}} \left[-1 + \frac{1}{\sigma^2} \int_0^\infty r e^{-\frac{1}{2}\frac{r^2}{\sigma^2}} J_0(kr)\, dr \right] \ .
\end{aligned} \tag{2.2.15}$$

Recognizing that the integral is a zero order Hankel transform (see Bracewell 1965), the equation reduces to

$$\begin{aligned}\frac{da_k(t)}{dt} &= \frac{2\nu a_k(t)}{\sigma^2}\left[e^{-k^2\sigma^2/2}-1\right]\\ &= -\nu\hat{k}^2 a_k(t)\ ,\end{aligned} \tag{2.2.16}$$

where

$$\hat{k}^2 = k^2\left(1-\frac{k^2\sigma^2}{4}+\frac{k^4\sigma^4}{24}-\cdots\right)\ . \tag{2.2.17}$$

$\hat{k}$ is a modified wave number and is a good approximation of k if the product $k\sigma$ is small. This is the case for wavelengths which are large compared to σ.

Applying a forward Euler time stepping to a solution of the type (2.2.10) will produce

$$\hat{a}_k{}^{n+1} = a_k^n + \Delta t\,\frac{d\hat{a}_k(t)}{dt}\ , \tag{2.2.18}$$

where $\frac{d\hat{a}(t)}{dt}$ is an approximation of the first time derivative that could be obtained using finite differences or, as in this case, with an integral representation of the Laplacian.

$$\hat{a}_k{}^{n+1} = a_k^n + \Delta t\left[-\frac{2\nu a_k^n}{\sigma^2}+\frac{2\nu a_k^n}{\sigma^2}e^{-\frac{k^2\sigma^2}{2}}\right]\ , \tag{2.2.19}$$

and if $\Delta t = \frac{\sigma^2}{2\nu}$, the previous equation simplifies to

$$\begin{aligned}\hat{a}_k{}^{n+1} &= a_k^n - a_k^n + a_k^n e^{-\nu k^2\Delta t}\\ &= a_k^n e^{-\nu k^2\Delta t}\end{aligned} \tag{2.2.20}$$

which is exact and valid for any k! The Gaussian smoothing operator is truly a smooth operator. When multiplied by the proper time step, the extra terms in the modified wave number are, in fact, the higher order time derivatives which have been neglected in the Euler time stepping.

This wasn't totally unexpected since the particle weights are always (for any Δt) redistributed according to a gaussian with a core size of σ. When the condition $\sigma^2 = 2\nu\Delta t$ is met, the numerical diffusion length, σ, is equal to the expected viscous diffusion length and the numerical solution should be exact. In the numerical experiment, the error doesn't completely vanish since the tail of the Gaussian has started to interact with the edges of the computational domain.

It should be noted that this "fortuitous" cancellation of the spatial error by the temporal error can only occur when a Gaussian kernel and a forward Euler time stepping are used. In this case, using a higher order time integration scheme will not result in a more accurate solution. The algebraic smoothing also shows some cancellation but applying the preceeding analysis to its associated diffusion kernel,

$$\eta_\sigma\left(|\mathbf{x}-\mathbf{y}|\right) = \frac{3}{4\pi\sigma^2}\frac{1}{\left(1+\frac{|\mathbf{x}-\mathbf{y}|^2}{4\sigma^2}\right)^4} \ , \tag{2.2.21}$$

it becomes clear that an exact solution is never achieved. It is found that

$$\begin{aligned} \frac{da_k(t)}{dt} &= \frac{2\nu a_k(t)}{\sigma^2}\left[k^3\sigma^3 K_3(2\sigma k) - 1\right] \\ &= -\nu\hat{k}^2 a_k(t) \ , \end{aligned} \tag{2.2.22}$$

but this time,

$$\begin{aligned} \hat{k}^2 &= k^2\left(1 - \frac{k^2\sigma^2}{4} + \frac{k^4\sigma^4}{6}\left(\frac{\psi(1)+\psi(4)}{2} - \ln(k\sigma)\right) + \cdots\right) \\ &= k^2\left(1 - \frac{k^2\sigma^2}{4} + \frac{k^4\sigma^4}{6}\left(\frac{11}{12} - \gamma - \ln(k\sigma)\right) + \cdots\right) \ , \end{aligned} \tag{2.2.23}$$

where $K_3(x)$ is a modified Bessel function, $\psi(x)$ is the digamma function and γ is the Euler constant. As in the case of the Gaussian regularization, the leading term of the modified wavenumber is correct and the WRM provides a valid approximation of the Laplacian. Again, the contribution of the second order time derivative is captured exactly when $\Delta t = \frac{\sigma^2}{2\nu}$ and in the region of maximum cancellation, the error is not a linear function of the time step anymore. However, with this particular choice of time step, the higher order time derivatives are not represented correctly and an exact solution cannot be obtained. Fig. (2.2.4) also shows that an even smaller error can be observed when a slightly bigger time step is used. That optimum time step minimizes a combination of higher order terms. However, it cannot be found analytically since the presence of the logarithmic term makes the cancellation wavenumber dependent. Consequently, the selection of a judicious time step for the algebraic smoothing is problem-dependent to the extent that different wavenumbers will dominate different problems. In the case of the diffusive Gaussian, most of the dissipation occurs at wavelengths comparable to the standard deviation of the distribution.

For both regularization functions, $\Delta t = \sigma^2/2\nu$ seems to be a very good choice. However, the same time increment is also used in the advective fractional step and a value that is optimal for viscous diffusion often turns out to be too large for convection, especially at low resolution. Besides, even if it were acceptable for both fractional steps, a smaller time step might still be needed to reduce the splitting error and insure a strong coupling between advection and diffusion.

2.2.3 Stability

Explicit time integration schemes applied to a diffusive process are prone to instabilities. If too large a time step is used, they are incapable of capturing the rapid decay of the short wavelengths and instead of being quickly dissipated, these modes oscillate wildly with an increasing amplitude. The shortest available wavelength depends on the refinement of the spatial discretization. The time step has to be selected in such a way that the amplitude of the shortest wavelength remains bounded for all time; these methods are conditionally stable. When a fine grid is used, the stability requirement can impose a very severe upper bound on the time step.

Larger time steps can be used with an implicit scheme since they can be constructed to be unconditionally stable. The drawbacks are that a matrix has to be inverted at every time step and that although stable, these methods are not necessarily accurate. Besides, one cannot design an implicit scheme in a Lagrangian context since the heat equation has to be written at the next time level. To do so, one needs the updated particle locations, which depend on the as yet undetermined circulations. When used with a vortex method, it seems that an explicit time stepping of the WRM is the only alternative and that the instabilities must be kept at bay by using a small enough time step.

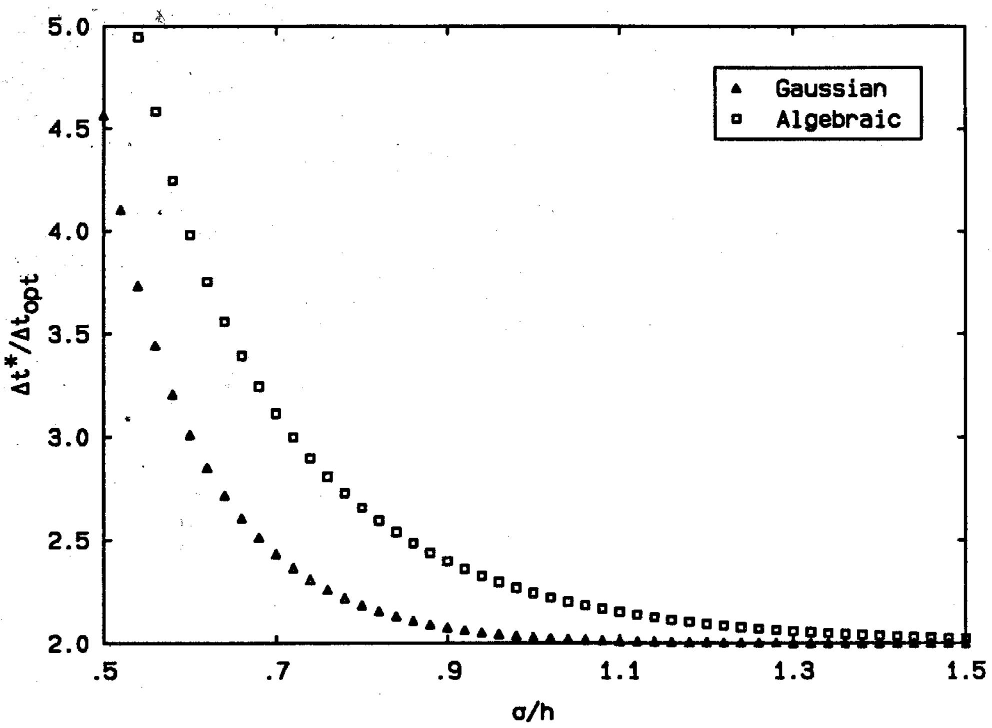


FIGURE 2.2.5 Limit of marginal stability as a function of the overlap.

In the general framework of a Lagrangian method, it is not possible to find the exact stability requirement since the particles are disorganized. A more tractable problem will be considered where the particles reside at the vertices of an infinite square mesh. When such a structure exists in the blob locations, a Von Neumann type analysis can be performed. If h is the distance between adjacent particles, the most unstable mode is the one associated with the Nyquist frequency for which $k_{\max} = \frac{\pi}{h}$. Assuming a perturbation in the x-direction,

$$a_{k_{\max}} e^{ik_{\max}x} \quad , \tag{2.2.24}$$

and a forward Euler time integration scheme, a time step Δt^* for which

$$a^{n+1}_{k_{\max}} = -a^{n}_{k_{\max}} \tag{2.2.25}$$

is sought. Again, the superscripts refer to the discrete time levels. The stability limit is plotted in Fig. (2.2.5) as a function of the overlap. The Euler time integration will be stable for any value of time step under the curves. In the limiting case $\frac{\sigma}{h} \to \infty$, the continuous WRM can be applied to the most unstable mode. Equations (2.2.16) and (2.2.22) can then be modified to show that $\Delta t^* = 2\Delta t_{\text{opt}}$ for both regularization functions. Since one would like to run at (or near) the optimum viscous time step, it is reassuring to see that this particular choice is stable for all overlap ratios.

However, Cottet (1989) pointed out that this simple-minded analysis is over-optimistic and the situation deteriorates when the Lagrangian mesh is deformed. The deformation produces smaller length scales that don't exist in the original mesh. This process will be modeled by uniformly compressing the mesh in the x-direction. To respect the divergence-free condition, the grid must expand in the y-direction as well.

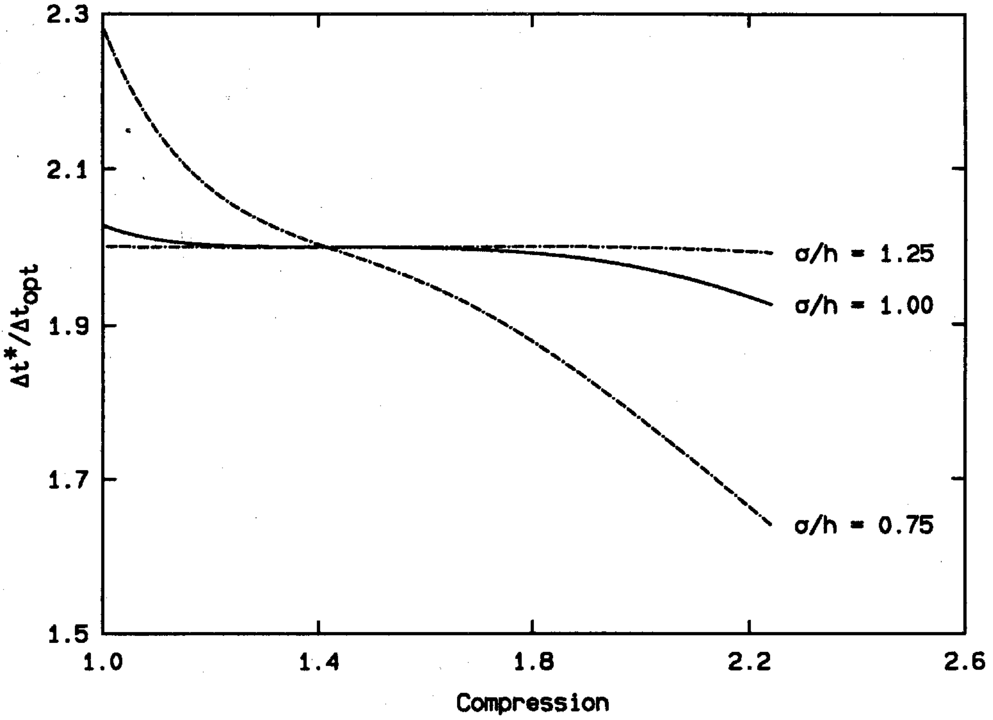


FIGURE 2.2.6 Effect of uniform compression on the stability limit (Gaussian regularization).

Figures (2.2.6) and (2.2.7) show the limit of marginal stability for reasonable values of the overlap as a function of the compression ratio. A compression of 2 means that the length scales have become 50% smaller in the x-direction. It can be observed that the stability margin is progressively reduced as smaller length scales are introduced. The method appears to be more robust when a large overlap is used, especially for the Gaussian smoothing. If the compression of the Lagrangian grid produces a more unstable scheme in the x-direction, it also has a negative effect in the y-direction. As the particles move apart, overlap is lost and so is the ability to represent a smooth function. Again, excessive overlap is a costly solution and in practice, remeshing (see Sec. 4.2.2) should be used to avoid the consequences of a deformed grid.

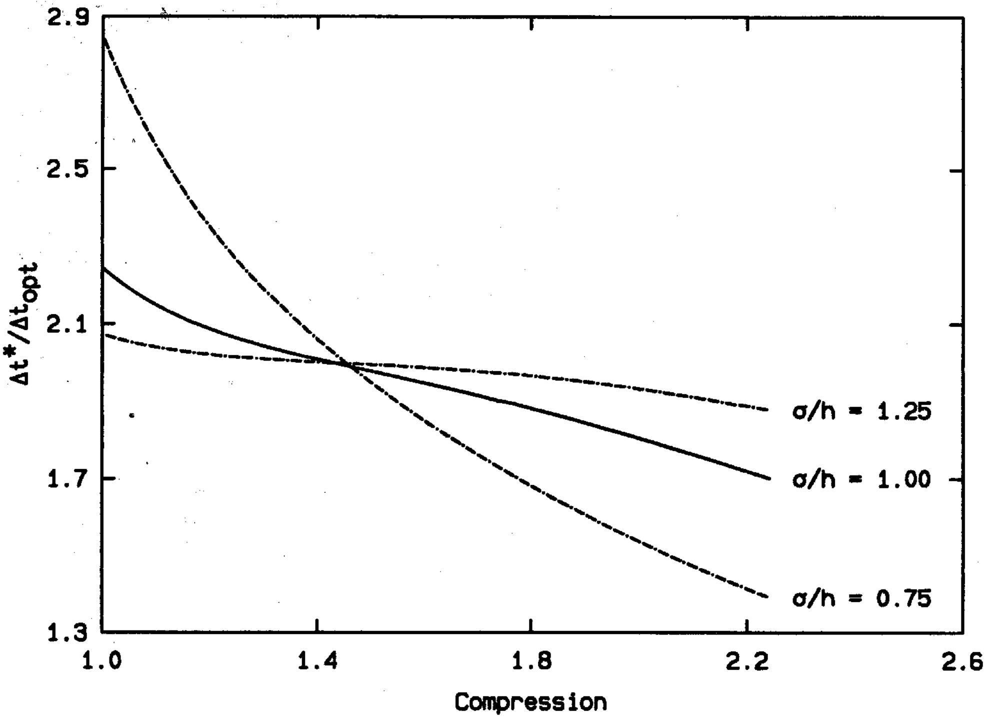


FIGURE 2.2.7 Effect of uniform compression on the stability limit (algebraic regularization).

2.3 Test Case

By simulating the viscous interaction of vortex rings at Re = 400, Winckelmans (1989) and Winckelmans & Leonard (1988) convincingly demonstrated that the WRM can handle the diffusion of vorticity in an infinite domain. The purpose of this section is to build confidence in the ability of that method to deal with solid boundaries. Being able to compare the numerical data with an exact solution will also be helpful in making the remaining numerical choices.

The test case should be simple enough to have an analytical solution but should also involve all the ingredients characteristic of a bluff body flow or, more specifically, the enforcement of the no-slip condition by the creation of vorticity at the walls, the diffusion of that vorticity from the walls to the fluid, and the subsequent convection of the vorticity by the fluid motion. However, it is not possible to find an analytical solution when all three ingredients are present. The third one will be dropped since the ability of vortex methods to adequately solve the convection equation has already been established.

The enforcement of the no-slip condition and the transfer of the newly created vorticity from the walls to the fluid will be tested by simulating the flow induced by a cylinder oscillating around its axis. The cylinder is immersed in a viscous fluid otherwise at rest and its motion is purely circumferential. The resulting velocity field is highly unstable and will be stabilized by enforcing axisymmetry. With this additional constraint, the convection term disappears and the problem becomes purely diffusive and stable.

Under that assumption, the radial momentum equation reduces to

$$\frac{\partial p}{\partial r} = \rho \frac{{v_\theta}^2}{r} \ . \tag{2.3.1}$$

The centrifugal acceleration is balanced by the radial pressure gradient. In the θ-direction, the conservation of momentum implies that

$$\frac{\partial v_\theta}{\partial t} = \nu \frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}(r v_\theta)\right) \ . \tag{2.3.2}$$

The cylinder has a unit radius and its motion is described by

$$v_\theta(r=1,t) = \Lambda \cos(\Omega t) \ . \tag{2.3.3}$$

Assuming a solution of the form

$$v_\theta(r,t) = \Re\{f(r)e^{i\Omega t}\} \ , \tag{2.3.4}$$

and defining

$$\beta = \sqrt{\frac{\Omega}{\nu}} \tag{2.3.5}$$

as a characteristic wavenumber, it can be shown that the circumferential velocity is

$$\begin{aligned} v_\theta(r,t) = {} & \frac{\Lambda \cos(\Omega t)}{k_\beta^2}\left[\mathrm{ker}_1(\beta)\,\mathrm{ker}_1(\beta r) + \mathrm{kei}_1(\beta)\,\mathrm{kei}_1(\beta r)\right] \\ & - \frac{\Lambda \sin(\Omega t)}{k_\beta^2}\left[\mathrm{ker}_1(\beta)\,\mathrm{kei}_1(\beta r) - \mathrm{kei}_1(\beta)\,\mathrm{ker}_1(\beta r)\right] \ , \end{aligned} \tag{2.3.6}$$

where $\mathrm{ker}_1(x)$ and $\mathrm{kei}_1(x)$ are Kelvin's functions of the first order (see Gray & MacRobert 1931) and

$$k_\beta^2 = \mathrm{ker}_1^{\,2}(\beta) + \mathrm{kei}_1^{\,2}(\beta) \ . \tag{2.3.7}$$

From that velocity field, one can easily determine the corresponding vorticity distribution,

$$\begin{aligned} \omega_z(r,t) = &\frac{\beta\Lambda}{\sqrt{2}k_\beta^2}\cos(\Omega t)\left[\mathrm{kei}_1(\beta)\, k^-(\beta r) - \mathrm{ker}_1(\beta)\, k^+(\beta r)\right] \\ &- \frac{\beta\Lambda}{\sqrt{2}k_\beta^2}\sin(\Omega t)\left[\mathrm{kei}_1(\beta)\, k^+(\beta r) - \mathrm{ker}_1(\beta)\, k^-(\beta r)\right] \ , \end{aligned} \tag{2.3.8}$$

where

$$k^\pm(\beta r) = \mathrm{ker}(\beta r) \pm \mathrm{kei}(\beta r) \ . \tag{2.3.9}$$

The moment, M_z, needed to maintain this oscillatory motion will be used as a diagnostic. For the axisymmetric case, the shear stress is related to the vorticity by

$$\tau_{r\theta}(r,t) = \omega_z(r,t) - \frac{2}{r}v_\theta(r,t) \ . \tag{2.3.10}$$

The moment is obtained by integrating the previous expression around the cylinder. Numerically, the moment on the body is estimated by differentiating the fluid angular momentum with respect to time and when a Gaussian regularization is used,

$$\begin{aligned} M_z = -\frac{dA}{dt} &= \frac{1}{2}\frac{d}{dt}\sum_i^N \alpha_i(r_i^2 + 2\sigma^2) + \pi R^3 \left.\frac{dv_\theta}{dt}\right|_R \\ &= \frac{1}{2}\frac{d}{dt}\sum_i^N \alpha_i(r_i^2 + 2\sigma^2) + \pi\Omega\Lambda\sin(\Omega t) \ . \end{aligned} \tag{2.3.11}$$

The second term appears as a result of the tangential acceleration of the wall. The summation is evaluated at every time step and a central difference scheme is used to differentiate it with respect to time and find its contribution to the moment.

To obtain a numerical solution, the fluid surrounding the cylinder is subdivided into cells of equal area. The cell generation scheme is similar to the one described in Christiansen (1973), except that the layers lying inside the cylinder have been removed. Even under the assumption of axisymmetry, the problem will be solved as if it were fully two-dimensional. However, since the fluid can only convect in a direction where there is no vorticity gradient, the particles will not be allowed to move. It should be noted that in this case, vorticity is created at the wall as a result of the tangential acceleration of the boundary and is not related to the motion of the vortices. The eight-fold symmetry in the grid will also be exploited to make the simulation more efficient.

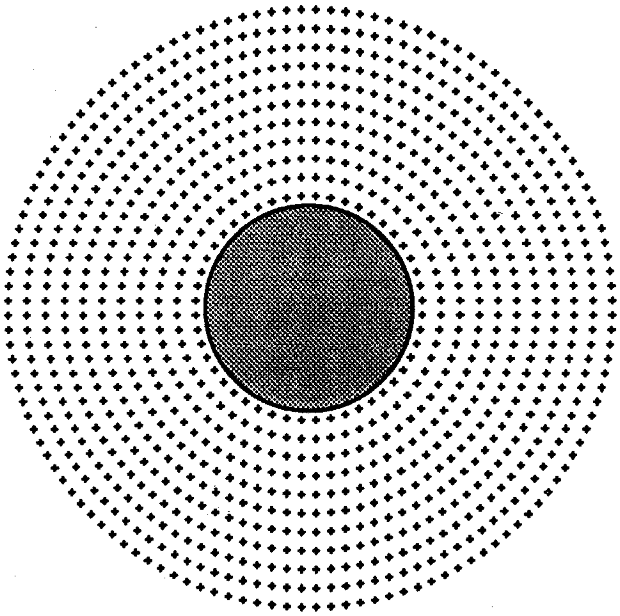


FIGURE 2.3.1 Distribution of the computational elements for the oscillating cylinder problem.

In the simulation that follows, the cylinder is surrounded by 15,312 particles distributed among 44 layers. Fig. (2.3.1) shows the distribution of the vortices around the cylinder (only one out of sixteen particles is actually shown on the figure). The region for which $r \geq 3$ is not discretized since it is expected to remain irrotational at all times. The cylinder reaches a maximum circumferential velocity, Λ, of one. Gaussian cores are used with an overlap, $\frac{\sigma}{\sqrt{A}} = 1$ and the time step is such that $\sigma^2 = 2\nu\Delta t$ which is the optimum choice as described in Sec. (2.2.2). The wavenumber β has been arbitrarily set to 10; a larger value would have produced stronger gradients and required a finer discretization.

Reaching the asymptotic solution from rest would be a waste of computer resources, especially if one wants to explore the parameter space. Instead, at $t = 0$ the particles are assigned a circulation corresponding to that asymptotic solution (2.3.8). The WRM (Eq. (2.1.18)) is used to update these circulations for one complete cycle of the cylinder motion. Once the cycle has been completed, the particle strengths should be back to their original values. This is found to be the case as demonstrated in Fig. (2.3.2). The numerical distribution of vorticity is nearly indistinguishable from the analytic solution and it is still the case if four more cycles are added to the simulation. This was achieved with a relatively coarse discretization as only 10 particles are located in the region where the gradient of vorticity is strong (between the wall and $r = 1.5$).

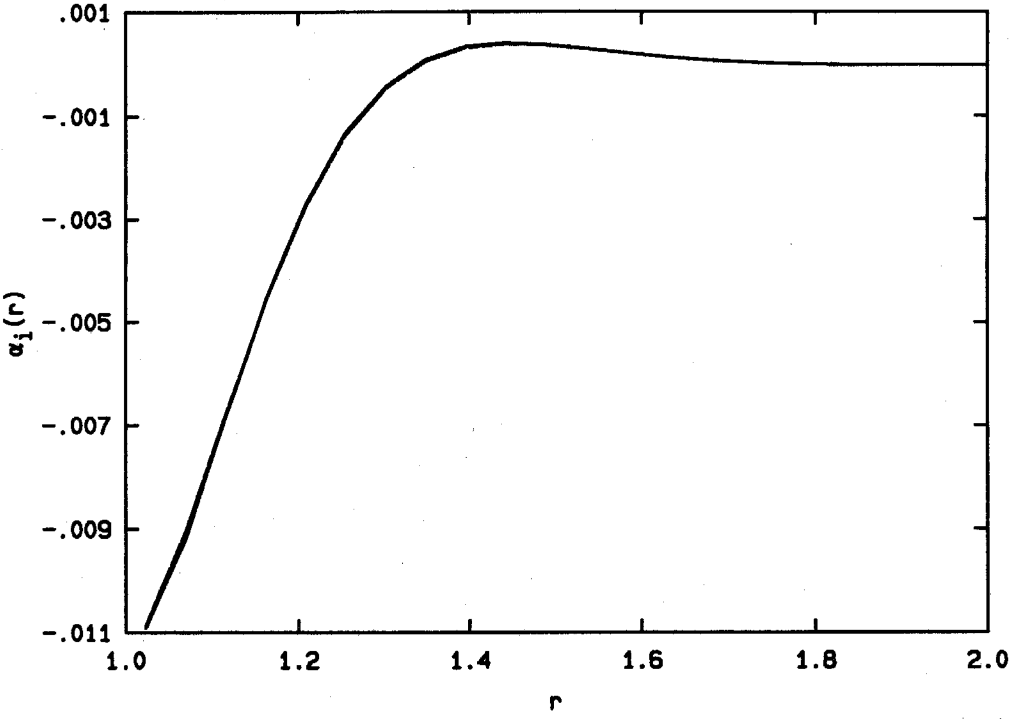


FIGURE 2.3.2 Comparison of the particle weights with the analytic solution (solid) after one (dash) and five cycles (dot-dash).

The computed moment also compares well with the analytical solution; both the sinusoidal nature and the amplitude of the moment time history have been captured by the numerical scheme. Fig. (2.3.3) shows a small phase shift between the two solutions but the magnitude of that shift can be reduced by using a finer discretization in space and time. In his vortex rings simulation, Winckelmans (1989) has shown that quadratic diagnostics are more sensitive to numerical errors than the linear ones. The computation of the moment involves the time derivative of the angular impulse, a quadratic quantity; so it should be easier to compute the drag properly since it is related to changes of the fluid linear impulse.

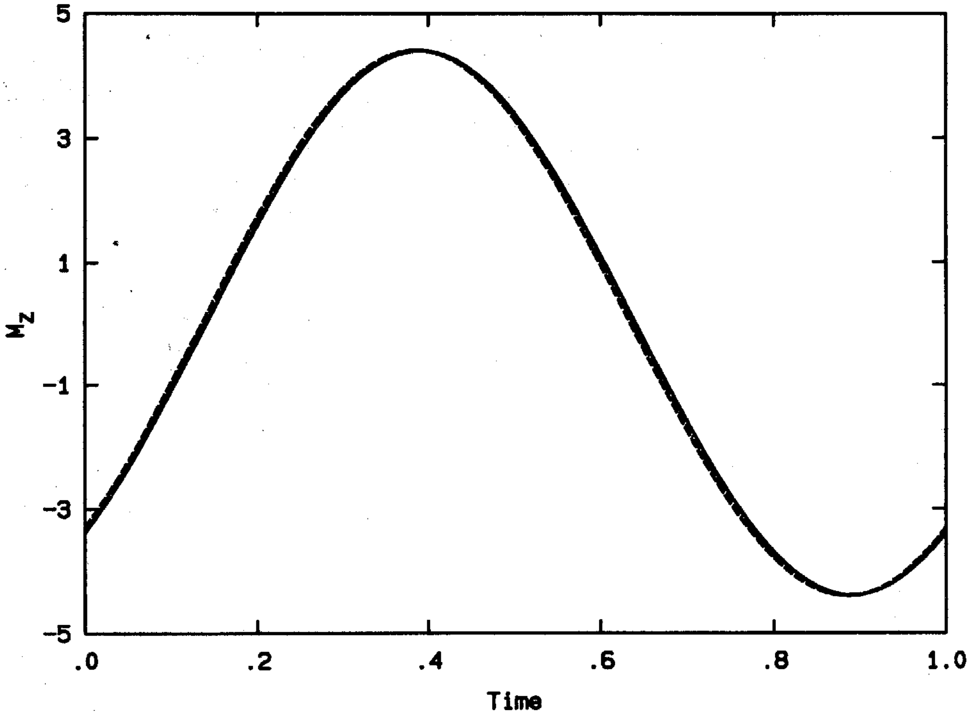


FIGURE 2.3.3 Moment on the cylinder obtained analytically (solid) and numerically for 44 (dash) and 62 (dot-dash) layers of particles.

All the above results were obtained with Gaussian cores and at least 250 straight panels to represent the cylinder. Calculations done with the algebraic smoothing were very similar but not quite as accurate and required slightly more computing time. Curved panels didn't improve the algebraic simulation as the large number of straight panels was already providing a good description of the cylinder surface.

Overall, the proposed numerical scheme did a fine job on this simple problem. To obtain similar results, a random walk approach would require a number of computational elements that far exceed the capability of existing computers. In practice, the cost associated with the convection step dictates the number of particles and, in that context, the slow convergence of the random walk scheme often results in noisy simulations. The next chapter presents a technique that reduces the cost associated with the advection of the vorticity field. The same computing resources then allow the use of a larger number of computational elements, and consequently, a better spatial resolution.

CHAPTER 3

A Fast Parallel Vortex Method

This chapter is mostly concerned with the cost of the velocity evaluation involved in inviscid vortex methods. These methods are generally limited by the speed of the computers, not by their memory. Using a classical approach, the user will exhaust his computer resources and/or run out of patience long before the storage capability of the computer has been exceeded. By computing the velocities at a faster pace, higher resolution can be obtained and more interesting problems can be considered. The next section presents a fast velocity solver based on multi-range approximations. It is shown that a significant speed-up over the classical approach is obtained by suitably approximating the interactions between distant groups of vortices. To do so, the proposed scheme combines Greengard & Rokhlin's expansions (1987) with Appel's Lagrangian data structure (1985). It appears that this combination is the most appropriate for parallel processing. The implementation of this fast algorithm on the Caltech-JPL MarkIII hypercube is discussed along with the use of the existing data structure to reduce the time required by the viscous interactions. Finally, the fast algorithm is extended to problems where a plane of symmetry requires the presence of image vortices.

3.1 Fast Algorithms

In order to move the vortex elements during the convective fractional step, the velocity must be evaluated at each particle location. A Green's function technique is used to solve

$$\nabla^2 \mathbf{u} = -\nabla \times (\omega \mathbf{e}_z) \ . \tag{3.1.1}$$

The velocity induced by an isolated point vortex,

$$\omega(\mathbf{x},t) = \alpha\,\delta(\mathbf{x}-\mathbf{x}_\alpha(t)) \ , \tag{3.1.2}$$

is

$$\mathbf{u}(\mathbf{x},t) = -\frac{\alpha}{2\pi}\frac{(\mathbf{x}-\mathbf{x}_\alpha(t))\times\mathbf{e}_z}{|\mathbf{x}-\mathbf{x}_\alpha(t)|^2} \ . \tag{3.1.3}$$

Alternatively, complex notation can be used and

$$\mathrm{w}(z,t) = u(z,t) + iv(z,t) = \frac{i\alpha}{2\pi}\frac{1}{(z-z_\alpha)^*} \ , \tag{3.1.4}$$

where z^* is the complex conjugate of z. When the vorticity field is regularized by $R_\sigma(z-z_\alpha)$, it is found that

$$\mathrm{w}(z,t) = \frac{i\alpha}{2\pi}\frac{g(|z-z_\alpha|/\sigma)}{(z-z_\alpha)^*} \ , \tag{3.1.5}$$

where

$$g(y) = 2\pi\int_0^y R(\zeta)\zeta\,d\zeta \ . \tag{3.1.6}$$

For a Gaussian regularization, smoothing effects are felt at a distance as far as 5 or 6 core radii from the center of the vortex blob. For a larger argument y, $g(y) \rightarrow 1$ and the vortex blob can be treated as a point vortex. In the following discussion of the fast algorithms, a singular description of the vorticity field will be assumed. This poses no limitations on the approach since these methods usually deal with the far-field of a particle. The next section describes how smoothing effects are reintroduced in a fast algorithm context.

Since Eq. (3.1.1) is linear, superposition is used to determine that the velocity field induced by

$$\omega(\mathbf{x},t) = \sum_j^N \alpha_j(t)\,\delta\,(\mathbf{x}-\mathbf{x}_j(t)) \tag{3.1.7}$$

is given by

$$\mathrm{w}(z,t) = \frac{i}{2\pi} \sum_{j}^{N} \frac{\alpha_j}{(z - z_j)^*} \quad . \tag{3.1.8}$$

The numerical scheme has transformed the original partial differential equation into a set of $2N$ ordinary differential equations: an N-body problem. This class of problems is encountered in many fields of computational physics, e.g., molecular dynamics, gravitational interactions, plasma physics, and of course, vortex dynamics. These problems involve N evaluations of a summation over $(N-1)$* interactions. Even if symmetry is used to cut the number of interactions by half, the resulting N^2 time complexity makes simulations using more than a few thousands particles prohibitively expensive.

3.1.1 Far-field approximations

When each pairwise interaction is considered, distant and nearby pairs of vortices are treated with the same care. As a result, a disproportionate amount of time is spent computing the influence of distant vortices that have little influence on the velocity of a given particle. This is not to say that the far-field is to be totally ignored since the accumulation of small contributions can have a significant effect. The key element in making the velocity evaluation faster is to approximate the influence of the far-field by considering groups of vortices instead of the individual vortices themselves. This fact was recognized by Spalart & Leonard (1981), in a different context by Appel (1985) and, as pointed out by Salmon (1990), probably by Newton himself! When the collective influence of a distant group of vortices is to be evaluated, the very accurate representation of the group provided by its vortices can be overlooked and a cruder description that retains only its most important features can be used. These would be the group location, circulation, and possibly, some coarse approximation of its shape and vorticity distribution.

A convenient approximate representation is based on multipole expansions. Consider a compact group of J point vortices,

---

* The self induced velocity is ignored for point vortices and is identically zero for regularized particles.

$$\omega_g = \sum_j^J \alpha_j \, \delta(z - z_j) \ , \tag{3.1.9}$$

where all vortices are located within a radius r_M of the group center, z_M. As discussed below, z_M is chosen to make the group as compact as possible. Other authors, like Appel (op. cit.), saw some benefits in locating z_M at the center of vorticity. For any choice of z_M, the vortices induce a velocity that can be expressed as

$$\begin{aligned} \mathrm{w}_g(z) &= \frac{i}{2\pi} \sum_j^J \frac{\alpha_j}{(z^* - z_j^*)} \\ &= \frac{i}{2\pi} \sum_j^J \frac{\alpha_j}{((z^* - z_M^*) - (z_j^* - z_M^*))} \ . \end{aligned} \tag{3.1.10}$$

Again, the asterisk refers to the complex conjugate of the quantity to which it is applied. Outside of the group, the velocity field can be rewritten as a multipole expansion,

$$\mathrm{w}_g(z) = \frac{i}{2\pi} \frac{1}{(z^* - z_M^*)} \left(a_0 + \frac{a_1}{(z^* - z_M^*)} + \frac{a_2}{(z^* - z_M^*)^2} + \cdots \right) \ , \tag{3.1.11}$$

where

$$a_k = \sum_j^J \alpha_j \, (z_j^* - z_M^*)^k \ . \tag{3.1.12}$$

In general, the coefficients a_k are complex numbers. However, a_0 is real, its imaginary part representing a source term.

In practice, the expansion is truncated to L terms and

$$\mathrm{w}_g(z) \simeq \frac{i}{2\pi} \frac{1}{(z^* - z_M^*)} \sum_{l=0}^{L-1} \frac{a_l}{(z^* - z_M^*)^l} \ , \tag{3.1.13}$$

which is valid for $|z - z_M| > r_M$. The contribution from the first neglected term drops like $1/(z - z_M)^L$. Therefore, even a truncated series will provide an accurate velocity estimate far from z_M.

It would be possible to build a fast algorithm at this stage by evaluating the multipole expansion at the location of particles that don't belong to the group. This is basically the scheme used by Barnes & Hut (1986). Greengard & Rokhlin (1987) went a step further by proposing group to group interactions. In this case, the multipole expansion is transformed into a Taylor series around the center of the second group, z_T, where the influence of the first one is sought. In the neighborhood of z_T, the induced velocity can be written as

$$\begin{aligned} \mathrm{w}_g(z) &= b_0 + b_1(z - z_T) + b_2(z - z_T)^2 + \cdots \\ &\simeq \sum_{l=0}^{L-1} b_p\,(z - z_T)^l \ , \end{aligned} \tag{3.1.14}$$

where

$$b_l = \left(\frac{-1}{z_T^* - z_M^*}\right)^l \sum_{k=0}^{L-1} \binom{k+l}{l} \frac{a_k}{(z_T^* - z_M^*)} \ . \tag{3.1.15}$$

An interaction between two groups consists of finding the coefficients of the Taylor series from the knowledge of the relative location of the groups and their respective multipole expansions. The work associated with this interaction is independent of the number of vortices in the groups. Consequently, the speedup over the N^2 approach is more interesting when large groups are involved. On the other hand, if the groups are small, it might be cheaper to consider all pairwise interactions between vortices. Assuming that the groups involved in the interaction have the same number of vortices, J, the critical J for which J^2 pairwise interactions of vortices require the same computational effort as one group to group interaction will be referred to as $J_{\min}$. No group with less than $J_{\min}$ vortices will be allowed since they would slow down the simulation. The threshold $J_{\min}$ is a function of L, the number of terms in the expansions. Since the work required to compute one group to group interaction is of order $\mathcal{O}(L^2)$, it might seem preferable to keep L to a minimum but then a larger error would result from each approximation. The error,

ϵ, is defined as the difference between the velocities obtained from a given group to group approximation and the ones resulting from all the pairwise interactions of the groups' members. Greengard & Rokhlin (op. cit.) have shown that when the same number of terms is kept in both expansions, ϵ is bounded by

$$\epsilon \leq A \left(\frac{r_{\max}}{\Delta r} \right)^{L} \quad , \tag{3.1.16}$$

where

$$\Delta r = |z_{M} - z_{T}| \quad , \tag{3.1.17}$$

$$r_{\max} = \max(r_{T}, r_{M}) \quad , \tag{3.1.18}$$

and

$$A = \sum_{j}^{J} |\alpha_j| \quad . \tag{3.1.19}$$

These three quantities are known, and an error estimate can be found for any pair of groups. If this estimate is smaller than some criterion, $\bar{\epsilon}$, the approximation is judged acceptable, and the computation can proceed with that group to group interaction**. If not, at least one of the groups is too large and the approximation is rejected since it would result in a significant error. In that case, the larger group is subdivided into two smaller ones, and an error estimate is found for the two new pairs of groups. If the error is still too large, the procedure is repeated until a valid approximation is found or until the smallest groups are reached. In the latter situation, pairwise interactions between vortices are used to determine the influence of one group on another.

** In practice, $(r_{\max}/\Delta r) \leq (\bar{\epsilon}/A)^{\frac{1}{L}}$ is cheaper to evaluate and is used instead of Eq. (3.1.16).

3.1.2 Data structure

An efficient implementation requires a data structure that is going to facilitate the search for acceptable approximations. As proposed by Appel, a binary tree is used. In that framework, a giant cluster sits on top of the data structure; it includes all the vortex particles. It stores all the information relevant to the group, i.e., its location, its radius and the coefficients of the multipole expansion. In addition, it carries the address of its two children, each of them responsible for approximately half of the vortices of the parent group. Whenever smaller groups are sought, these pointers are used to rapidly access the relevant information. The children carry the description of their own group of vortices and are themselves pointing at two smaller groups, their own children, the grand-children of the patriarchal group. More subgroups are created by equally dividing the vortices of the parent groups along the "x" and "y" axis alternatively. This splitting process stops when all groups have approximately $J_{\min}$ vortices. Then, instead of pointing toward two smaller groups, the parent node points toward a list of vortices. As shown in Fig. (3.1.1), the data structure provides a quick way to access groups, from the largest to the smallest ones, and ultimately to the individual vortices themselves.

3.1.3 Velocity evaluations

Once the groups have been identified and hierarchically ordered, the coefficients of the multipole expansion that will represent everyone of them need to be evaluated. Having access to the vortices belonging to every group, Eq. (3.1.12) could be used for this purpose but it would be costly, especially for the larger groups. Eq. (3.1.12) is only used to find the coefficients of the smallest groups in the data structure, the ones that have direct access to the vortices. Then, the coefficients of the children are used to find the multipole expansion of their parent group. The expansions are constructed from the bottom up. The coefficients of the left child adequately describe its content with respect to the center of its group, z_M. To represent the left half of the parent node, that multipole expansion has to be shifted to the center of the parent node, $z_M{}'$ and the new coefficients are:

$$a'_p = \sum_{p=0}^{k} \binom{p}{k-p} a_p (z^*_M - z^*_M{}')^{k-p} \ . \tag{3.1.20}$$

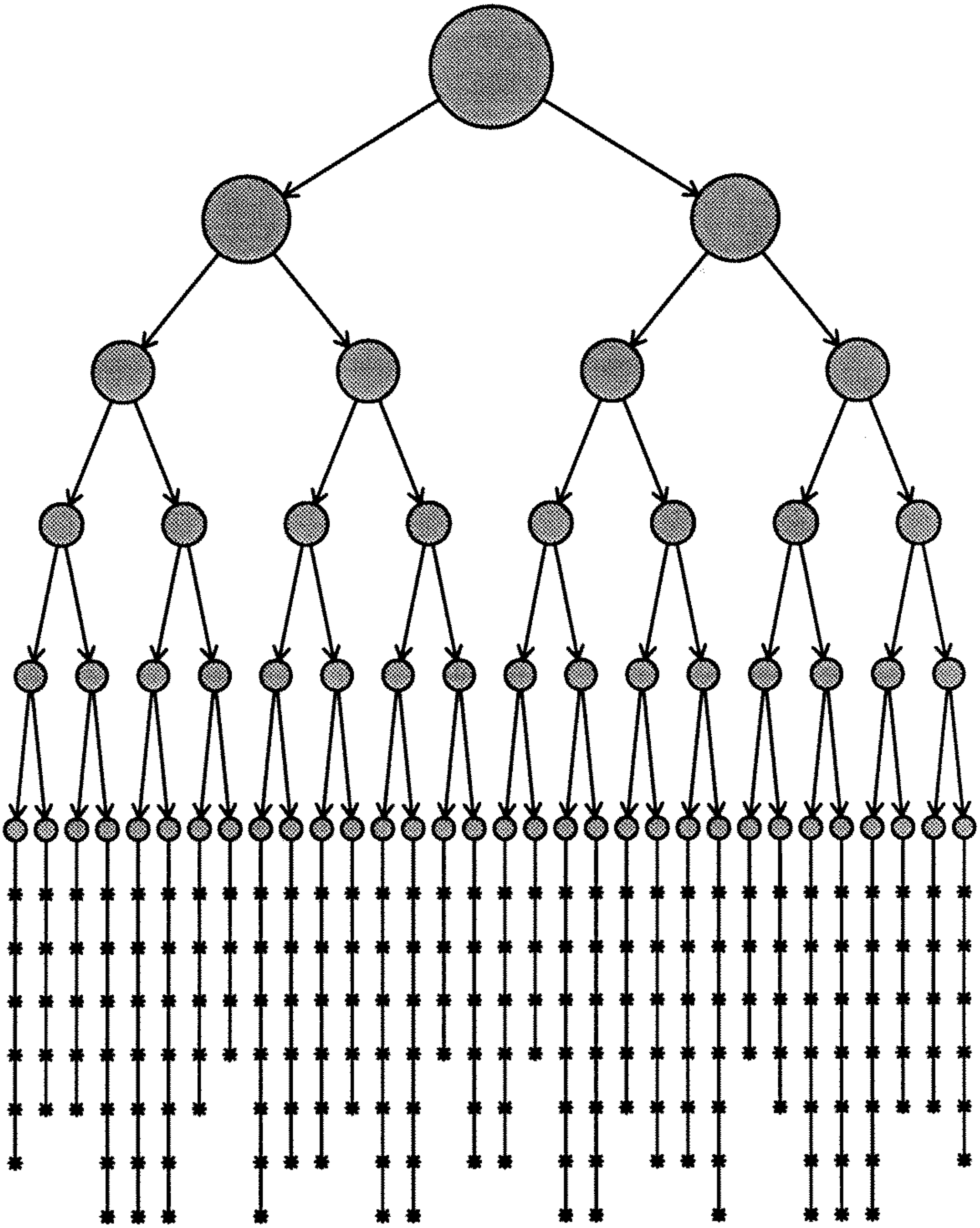


FIGURE 3.1.1 Fast algorithm's data structure

The same operation is repeated for the right child and its shifted coefficients are added to the ones of the left child to form the multipole expansion of the parent group. Recursive subroutines are used to repeat this assembling process until the top of the binary tree is reached.

Once the data structure is ready, the velocity evaluations can take place. The search for suitable pairs of groups is done with the help of recursive subroutines. **within**() and **between**(), similar to the ones used by Appel. The subroutine **between**() finds the influence of one group on another while **within**() computes the velocities within a group. It does so by finding the interaction between its left and right halves, after which the subroutine calls itself to compute the interactions within each half. Schematically, it looks like:

```
within(parent)
{
 between(left,right)
 within(left)
 within(right)
}
```

A **within**() of an indivisible group is simply the N^2 interaction of all its members. The interaction between two groups can be written as

```
between(left,right)
{
  if ( $\epsilon$ acceptable )
    compute(left,right)
  else
    if ($r_{\text{left}} > r_{\text{right}}$)
      between(left of left,right)
      between(right of left,right)
    else
      between(left,left of right)
      between(left,right of right)
}.
```

If the error estimate associated with that pair of groups is acceptable, **compute**() is called to update the Taylor coefficients of each group. If this approximation is rejected, the largest group is split into two parts, and each half interacts with the group that was not subdivided. The subroutine calls itself with smaller and smaller groups until the error estimate is small enough or until the groups cannot be subdivided anymore. In the latter case, **between**() does not check the error estimate but immediately proceeds with the pairwise interaction of the vortices. Either alternative concludes the interaction of the two groups involved in the last call to **between**() which returns to the subroutine that called it. Before the original call to **between**() returns, all the **between** subroutines called in the process must return as well. For the user, it appears that all velocities are computed by a single call to **within**(top), then the two subroutines will call themselves thousands of times until all interactions have been accounted for.

At the end of this process, some of the velocities have been directly assigned to the individual vortices but most of the information about the velocity field lies in the Taylor coefficients of the groups. To update the location of the particles, the information accumulated in these coefficients has to be transferred downward to the appropriate vortices. The Taylor series of each group could be evaluated at all the appropriate locations but instead, shifting operations are used again. This procedure is similar to the one that took place to find the multipole coefficients with the distinction that it proceeds from the top to the bottom of the data structure. The Taylor series of the parent groups, centered around z_T, are systematically shifted toward the center of their children group, z'_T. The shifted coefficients are

$$b'_p = \sum_{\substack{k \\ k \geq p}}^{L} \binom{k}{p} (z'_T - z_T)^{k-p} \tag{3.1.21}$$

and are simply added to the existing ones. After they have received the contribution from their parent node, the updated coefficients are shifted downward to their own children. The process stops when the bottom of the data structure is reached; the Taylor series of the smallest groups are then evaluated at each particle location. Greengard & Rokhlin have shown that the error estimate of Eq. (3.1.16) is not affected by these shifting operations. At this point, the velocity of each vortex blob is known and an ODE solver is used to update its location. New multipole expansions are built from the new locations, and the next velocity evaluation can take place.

Appel's data structure is Lagrangian since it is built on top of the vortices and moves with them. It can be used for many time steps, but eventually, the groups will deform and could even begin to overlap. They would not be as compact as the original groups and the fast algorithm performance would deteriorate. To prevent this, the original data structure is discarded every few time steps (10 is a typical number) and new groups are identified from scratch by alternatively dividing the vortices along the "x" and "y" axis.

The data structure used by Greengard & Rokhlin is based on a spatial decomposition of the computational domain and consequently, has an Eulerian nature. The domain is subdivided into four square cells of equal area. The cells that contain more than $J_{\min}$ vortices are subdivided again and so forth. As the vortices move, they have to be sorted again in this set of rigid boxes. This step requires little work but complicates a parallel implementation as vortices have to be exchanged between processors after each time step. On the other hand, the acceptable pairs of groups are known a priori when when a rigid data structure is used and a parallel implementation can benefit from this predictability (see Katzenelson 1989). For the same reason, the Greengard & Rokhlin algorithm is also a better candidate for vectorization.

3.1.4 Fast algorithm performance

To evaluate the performance of the fast algorithm, velocities are computed for N vortices randomly distributed over a 1×1 square computational domain; their circulation is also assigned randomly. For fast algorithms based on multi-range approximations, this problem is actually a worse case scenario. When the vortex blobs are spread nearly uniformly, the groups have to be created artificially and cannot be as compact as the ones obtained in a problem where the vortices are naturally clustered.

The velocities are first computed to double precision accuracy with the N^2 method. This is considered as the exact solution and is used as a reference against which the approximate velocities can be compared. The combination of L and $\bar{\epsilon}$ are chosen in such a way that results obtained with the fast algorithm are indistinguishable from a single precision accuracy N^2 simulation. This is a very severe restriction since the numerical integration of these velocities in time is certainly not accurate to one part in a million. However, as pointed out by Barnes & Hut, the error due to the group to group approximations could accumulate over many time steps so that one cannot allow too large an error at any given time step. In the proposed scheme, the same data structure is used for many time steps and as a result, the error vectors are correlated over a few time steps. The severe restriction on $\bar{\epsilon}$ is needed to make the presence of the fast algorithm as inconspicuous as possible.

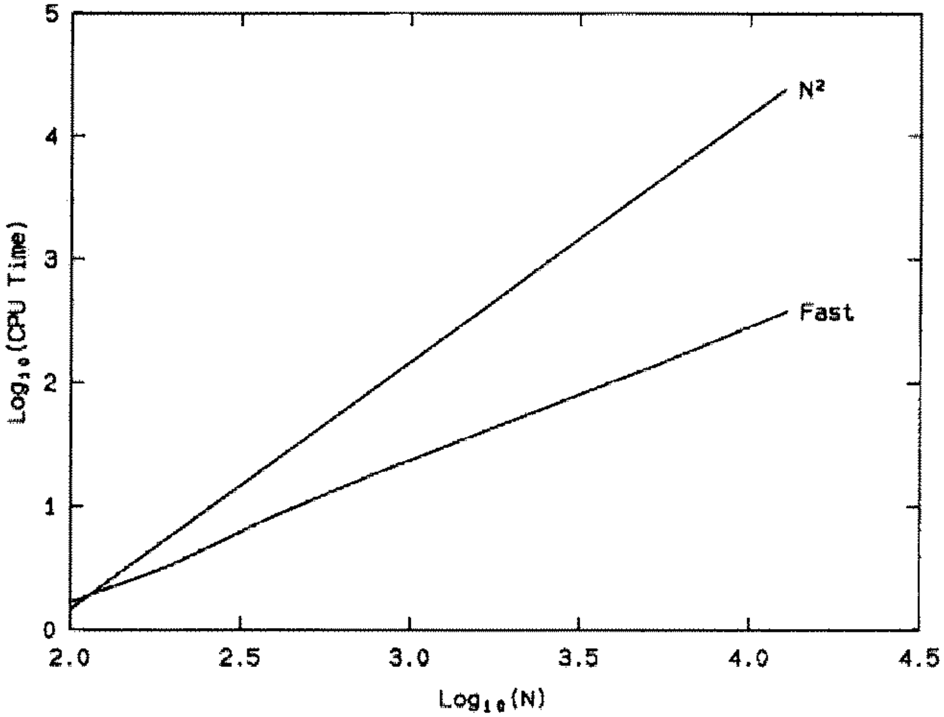


FIGURE 3.1.2 Performance of the fast algorithm

Despite this severe requirement, a remarkable speed-up over the classical approach can be observed in Fig. (3.1.2)†. The crossover occurs for as few as 150

---

† The CPU times are expressed in seconds on a VAX 750.

vortices; at this point, the extra cost of maintaining the data structure is balanced by the savings associated with the approximate treatment of the far-field. When N is increased further, the savings outweigh the extra bookkeeping and the proposed algorithm is faster than its competitor by a margin that increases with the number of vortices.

If it is clear that the computer requirement of the classical approach grows like the square of the number of vortices, it is not as simple to determine the growth rate for the fast algorithm. Because it only involves group to vortex interactions, the Barnes & Hut scheme can be shown to be $\mathcal{O}(N \log N)$. This analysis is based on the fact that any given particle interacts with M particles in its near-field and $\log N$ groups in its far-field. Since M doesn't depend on the total number of particles, the time complexity is indeed $\mathcal{O}(N \log N)$. Group to group interactions, such as presented here, remove some redundancy present in the Barnes & Hut scheme but at the same time, prevent an analysis based on the behavior of individual particles. While allowing these interactions, Greengard & Rokhlin used their rigid data structure to determine an upper bound to the number of floating point operations. It was determined that their algorithm is actually $\mathcal{O}(N)$. In the proposed algorithm, the flexible data structure prevents that systematic operation count and the time complexity cannot be determined analytically. Appel also used group to group interactions with a flexible data structure and applied a Barnes & Hut type of argument to claim an $\mathcal{O}(N \log N)$ time complexity. It would have been more appropriate to claim that, as in the proposed algorithm, the time complexity is at most $\mathcal{O}(N \log N)$. The two decades worth of data shown in Fig. (3.1.2) are not enough to determine the time complexity "experimentally." From this, one can only conclude that the difference between $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ makes very little difference in practice.

What is really important is the constant multiplying the leading order term. It was determined that the proportionality constant of the Barnes & Hut algorithm applied to a two dimensional vortex method is approximately 2.5 times larger than the constant of the proposed algorithm. This is the result of redundant decision making present in the former scheme. Nearby vortices should interact with the same distant groups but since they are sequentially compared to the binary tree, nearly identical error estimates have to be computed over and over. The proposed scheme eliminates that redundancy and also exploits the symmetry involved in group to group interactions. On the other hand, the simplicity of the Barnes & Hut algorithm is an advantage when the code is transported to concurrent computers or extended to three dimensional vortex methods.

The proportionality constant depends on both the required accuracy and the number of terms kept in the expansions. For the prescribed single precision accuracy, $L = 5$ was found to be the optimum choice and the results shown on Fig. (3.1.2) were obtained with this value. Keeping more terms allows interactions between larger groups, but these additional savings could not recoup the extra cost associated with each group to group approximation. On the other hand, the performance of low order schemes was not satisfactory either. For $L = 1$, it took thousands of vortices before the "fast" algorithm became competitive with the classical N^2 approach. On the other hand, Appel's scheme did a fine job in the astrophysical application while keeping only one term[‡]. It should be noted that his method was applied not only to a different problem, but more importantly, to a different kernel. Specifically, the gravitational interaction of galaxies involves a $(\frac{1}{r^2})$ kernel instead of the $(\frac{1}{r})$ Biot-Savart kernel for two dimensional vortical flows. The problem is more localized, the influence of the neglected terms drops more rapidly, and as a result, one or two terms are enough. The Biot-Savart kernel for a three dimensional vortex method is also $(\frac{1}{r^2})$, and low order schemes are probably appropriate as well.

Greengard & Rokhlin kept 20 terms in their expansions. By doing so, they could allow interactions of large groups even when they were fairly close. Actually, since every non-adjacent pair of square boxes can interact, the error resulting from a $r_{\max}/dr = 0.35$ has to be acceptable. Another reason why their scheme needs a larger L is that the center of the box is not necessarily the best center of expansion for the vortices inside that box. Since the error resulting from an approximation is a strong function of the group radius, it is important to build the data structure in such a way as to minimize the radii. In the proposed scheme, when two groups are assembled to form a larger one, the center of the parent group is chosen to satisfy this criterion. On the other hand, Greengard & Rokhlin could exploit the regular

---

[‡] Two terms are kept in the multipole expansions but only one in the Taylor series; so $L = 1$.

patterns of their data structure to accelerate the shifting operations. Even if the speed-up that they obtained is comparable to the one presented here, it is believed that a method that results in a lighter data structure is a superior alternative. This is not really a factor on a sequential machine since even a fast vortex method is still limited by speed, not by the memory requirement. But in a parallel implementation where portions of the data structure are to be exchanged between processors, it is crucial to describe the groups with as few coefficients as possible. In essence, fast algorithms trade memory for speed; a faster velocity evaluation is made possible by the additional information stored on top of the vortices. The memory required to store the data structure grows like $\mathcal{O}(NL)$. For $L = 5$ and the corresponding $J_{\min}$, the proportionality constant is such that the binary tree occupies approximately the same space as the N vortices.

The use of recursive subroutines to search through the binary tree for acceptable interactions does not lend itself to vectorization. However, it is still true that the interactions are independent events. The influence of `A` on `B`, where `A` and `B` can be either vortices or groups of vortices, can be determined without any regard to the vorticity field that surrounds them. That inherent parallelism can be exploited to implement the method on concurrent processors. The parallel version of the algorithm is discussed in Sec. 3.3. In the next sections, the fast algorithm will be extended to regularized vorticity fields and it will be shown how the diffusive fractional step can benefit from the binary tree data structure.

3.2 Smoothing and Viscous effects

The fast algorithm described in the previous section was derived for a singular vorticity distribution but it can be extended to smooth vorticity fields as well. In doing so, it is essential to recognize that the smoothing effects of the regularization kernel are only noticeable in the immediate vicinity of a particle. A smoothing radius, r_σ, can be defined outside of which a regularized particle can be treated as a point vortex; typically, $r_\sigma = 5\sigma$ where σ is the core size of the vortex blob. Smoothing is essentially a near-field phenomenon and can be included in fast algorithms by noticing that the near and the far-field are treated differently. The influence of the far-field is captured by group to group interactions and no regularization can be allowed at that level. However, a particle interacts with its near-field by pairwise interactions with neighboring vortices and the singular behavior of the vorticity field is no longer necessary. Smoothing effects can be included when the bottom of the tree is reached and the method shifts back to vortex to vortex interactions. One has to insure that the effective near-field of the fast algorithm extends to at least r_σ from any particle. This can be done by using a larger $J_{\min}$ or by adding an additional restriction on group to group interactions. In the latter case, one simply has to make sure that any vortex blob belonging to group A is at least r_σ away from all members of group B. A group to group approximation is then acceptable if $(dr - r_A - r_B) > r_\sigma$, in addition to the usual constraint on the error estimate.

Since viscous effects are also a local phenomenon, a viscous radius, r_ν, is defined along the same lines as r_σ. For a Gaussian regularization, r_ν is actually equal to r_σ. At a distance greater than r_ν from any particle, the heat kernel is small enough so that the viscous interactions can be ignored. Since there are no long-range viscous interactions, the data structure is used to eliminate the far-field of any given particle. Again, recursive subroutines are used. In the viscous counterpart of **between**, the quantity $(dr - r_A - r_B)$ is compared with r_ν. If this quantity is larger than r_ν, no viscous interactions take place between the vortex blobs of these two groups; the subroutine returns immediately. If the two groups come within r_ν of each other, the procedure is repeated with smaller groups. The method attempts to eliminate as many interactions as possible and when indivisible groups are reached, their vortex blobs interact viscously. Finally, diffusion also takes place within every indivisible group. Without any modification, the existing data structure was used to identify the nearest neighbors of every particles and, by doing so, to significantly speed up the viscous fractional step. In the absence of solid boundaries, the time spent for that fractional step grows like $\mathcal{O}(N)$.

3.3 Parallel implementation

The fast algorithm discussed in the previous sections was implemented on the Caltech-JPL MarkIII hypercube. This Motorola 68020-based multi-processor has 4 Megabytes of memory per node. Up to 128 processors can be connected in an hypercube topology. The MarkIII multicomputer is a MIMD (Multiple Instruction Multiple Data) machine, which means that the individual processors independently work on different streams of data.

To quantify the quality of the parallel implementation, the concurrent efficiency, E, is defined as

$$E = \frac{S}{P} \tag{3.3.1}$$

where P is the number of processors and S is the speed-up obtained over the same application running on a single processor. The efficiency, E, is always smaller than one because the parallel implementation involves operations not present in the sequential code such as communications between processors. Furthermore, the parallel code requires some duplication of the data and the merging of informations coming from different processors. Another source of inefficiency, or overhead, is load imbalance. If the work load is not distributed equally, a processor with a light assignment has to wait for its busier colleagues. This idle time, absent in a sequential code, adversely affects the speed-up, S.

While load imbalance dominates the overhead for the concurrent fast algorithm, it is not a problem for the parallel N^2 method which is known to be very efficient (see Fox, Johnson et al. 1988). In that framework, since any pair of vortices represents the same amount of work, the load can be perfectly balanced by assigning the same number of vortices to each processor. Furthermore, the domain decomposition can be done without paying any attention to the location of the vortices. To find the velocities, each processor makes a copy of its vortices and sends it to half of the other processors where it interacts with the resident vortices. The contributions to the velocities of the visiting copy are accumulated as it is sent from processor to processor. Ultimately, it is sent back to its original processor where these contributions are added to those of the copy that stayed there. A large amount of data has to be exchanged between processors. However, this application is so intensive that the time spent computing velocities dwarfs the communication time and efficiencies close to unity can be achieved for large problems. The regularity of the problem also allows a synchronous implementation (under CrOS) which further reduces the time spent communicating between the nodes

The global nature of the N^2 approach has made its parallel implementation fairly straightforward. However, that character was drastically changed by the fast algorithm as it introduced a strong component of locality. Globality is still present since the influence of particle is felt throughout the domain, but more care and computational effort is given to its near-field. The fast parallel algorithm has to reflect that dual nature, otherwise an efficient implementation will never be obtained. Moreover, the domain decomposition can no longer ignore the spatial distribution of the vortices. Nearby vortices are strongly coupled computationally. Hence, it makes sense to assign them to the same processor. The binary tree data structure could be used for that purpose. By dismissing the $(P-1)$ largest groups in the data structure, P groups containing approximately the same number of vortex blobs can be identified and a different processor is assigned the responsibility of each of these subtrees. For example, Fig. (3.3.2) shows the portion of the data structure assigned to processor 1 in a four processor environment. Obviously, more sub-trees can be created when the number of processors is larger than four. This strategy ensures that the vortices given to each processor are actually neighbors in the physical space. The drawback of this approach is that the full data structure has to be constructed in the host processor before portions of it can be sent to the hypercube. In practice, binary bisection is used in the host to spatially decompose the domain. Then, only the vortices are sent to the processors where a binary tree is locally built on top of them. Less data has to be loaded on the hypercube and the generation of the local binary trees can be done in parallel.

However, sending a copy of local data structure resulting from this domain decomposition to half the other processors does not necessarily result in a load balanced implementation since the work associated with processor to processor in-

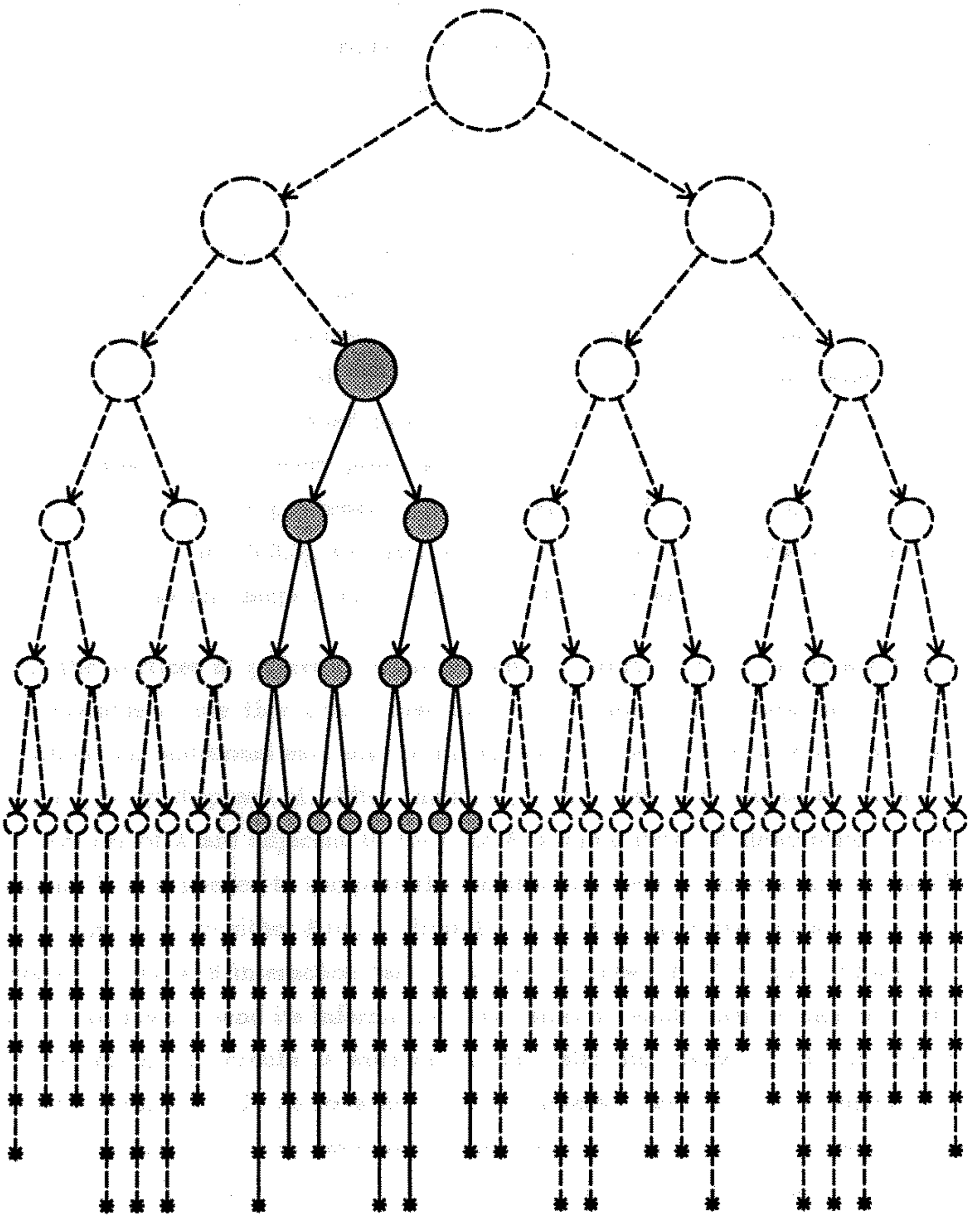


FIGURE 3.3.2 Data structure assigned to processor 1

teractions now depends on their respective location in physical space. Besides, a processor whose vortices are located at the center of the domain is involved in more costly interactions than a peripheral processor. To achieve the best possible load balancing, that central processor could send a copy of its data to more than half of the other processors and hence, be itself responsible for a smaller fraction of the work associated with its vortices.

Before a decision is made on which one is going to visit and which one is going to receive, the number of pairs of processors that need to exchange their data structure needs to be minimized. Following the domain decomposition, the portion of the data structure that sits above the subtrees is not present anywhere in the hypercube. That gap is filled using recursive doubling to make the description of the largest group of every processor known to everybody else. By limiting the broadcast to one group per processor, a small amount of data is actually exchanged but, as seen on Fig. (3.3.3), this step gives every processor a coarse description of its surroundings and helps it find its place in the universe.

If the vortices of processor `A` are far enough from those of processor `B`, it is even possible to use that coarse description to compute the interaction of `A` and `B` without an additional exchange of information. The far-field of every processor can be quickly disposed of. After thinking globally, one now has to act locally; if the vortices of `A` are adjacent to those of `B`, a more detailed description of their vorticity field is needed to compute their mutual influence. This requires a transfer of information from either `A` to `B` or from `B` to `A`. In the latter case, most of the work involved in the `A-B` interaction takes place in processor `A`. Obviously, processor `B` should not always send its information away since it would then remain idle while the rest of the hypercube is working. Load balancing concerns will dictate the flow of information. To do so, a list of all the interactions requiring a further data exchange is drawn in every processor. Since the upper portion of the tree has been duplicated P times, an identical copy of that list is created simultaneously in every processor. Then the responsibility of each item in the list is assigned to either processors involved while trying to distribute the resulting computational load as equally as possible.

Since vortices move only slightly during each time step, the computational work

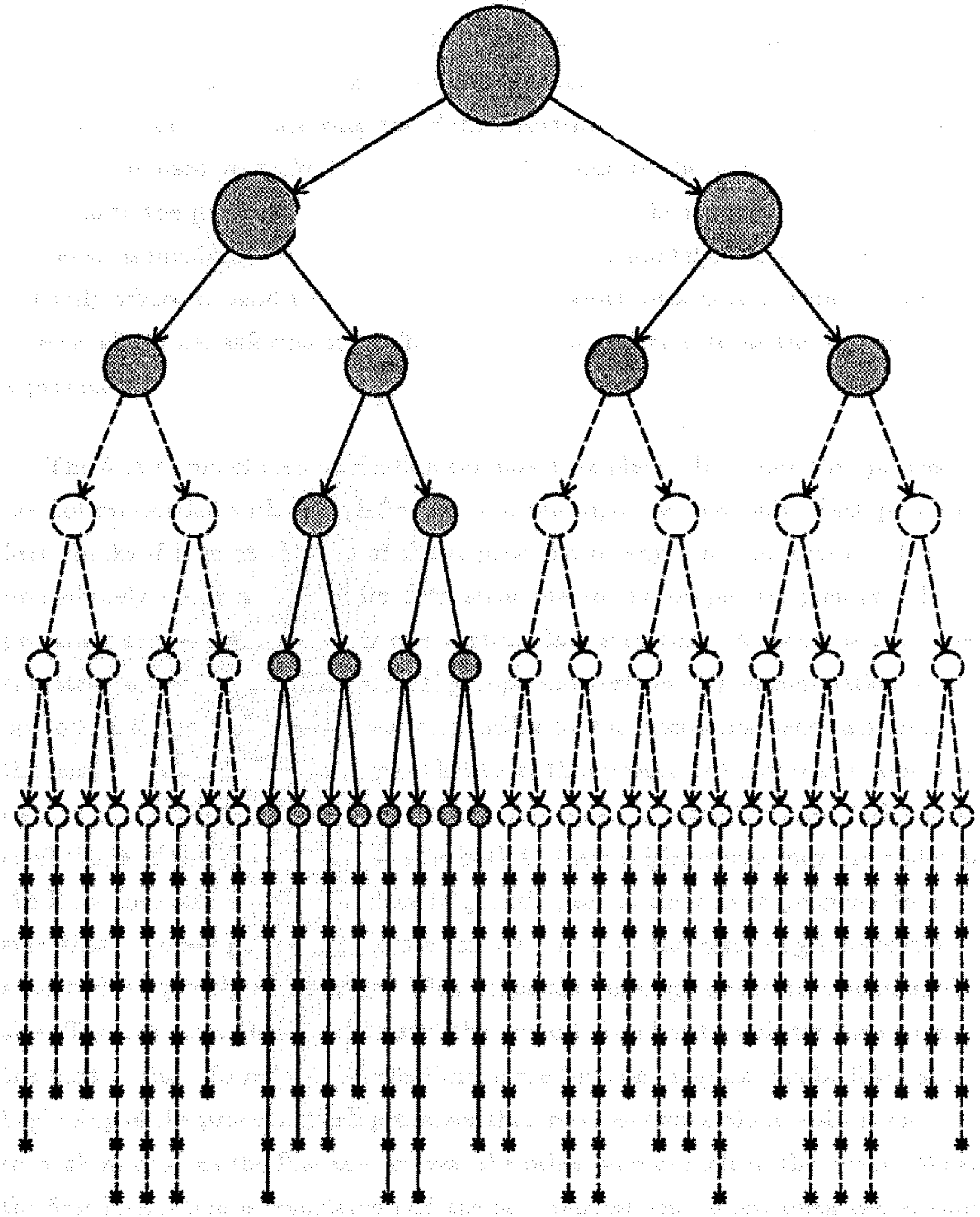


FIGURE 3.3.3 Data structure known to processor 1 after broadcast

required for the interaction of two given processors at the previous time step can be used as an estimate of the work involved for the present one. The pairs in the list are examined sequentially; the processor with the lightest work load when the pair is considered is given the responsibility of the interaction and computes the interaction after receiving the data structure from the other one. The work load that is used to make that decision is the sum of the work estimates already assigned to the processor plus half of the estimates of the interactions in which that processor is involved but have yet to be assigned. Ultimately, every processor knows not only where to send its data but also from which processor it should expect to receive additional information. The latter will be referred to as the request list of a processor.

The first round of communication can now take place. To ensure that processors are not overloaded with data, information is sent upon request only. Each processor first checks if it is at the top of the request list of any other processors. If so, it immediately sends a copy of its data structure to the proper recipient(s). Every processor receives one and only one visiting data structure. As soon as it arrives, this structure interacts with the local groups and vortices. Upon completion of that operation, the processor which was responsible for the interaction sends a message to the next processor in its own request list to let that processor know that a copy of its data is now needed at a specific location. The updated velocities and Taylor series coefficients of the visitor are also sent back to their origin where they are added to the local data structure. Processors frequently peek at their message queue to make sure that requests get an immediate answer and that the returning information is absorbed as quickly as possible. This keeps the message queue to a manageable size. The processor that has just sent the request then has to wait for the arrival of the next visitor. To reduce the idle time, more than one request can be filled at the beginning of the process. Each processor then receives two or three visitors and gets to work as soon as the first one arrives; the other ones are left in the queue. When the first interaction is completed and the next request sent, a processor can already start working on the next visitor in the message queue. It would be desirable to use as large a queue as possible, but memory restrictions limit its size to two or three visiting data structures. When all visiting copies have returned to their origin, the processors consider the interactions among their own vortices. Then, the vortices location and the whole data structure are updated. The process starts over again by broadcasting the largest group of each processor. Obviously, this message sending takes place asynchronously. Furthermore, the MarkIII is considered as a collection of computers loosely connected through an arbitrary network; the hypercube topology is not used as such.

At this point, it should be noted that the data structure used in a parallel implementation differs significantly from the one used on a sequential computer. In the latter case, the parent group points toward his children using memory addresses. On concurrent computers, the local binary trees are exchanged between processors and addresses that were valid where the tree was constructed are meaningless in a different processor. Instead, the data structure is built inside a one dimensional array and parent groups refer to their children by their indices. Two arrays are actually used, one for the vortices, **V**[ ], and one for the groups, **G**[ ]. When additional information is requested, **G**[ ] is sent immediately; then the respective location of the processors vortices is considered to determine if **V**[ ] should follow. If the processors are adjacent, the full description of the vorticity field is needed but if they are sufficiently far away, the description provided by the groups is adequate and **V**[ ] can stay home.

The computation of symmetric flows by a vortex method involves the presence of image vortices. Before actual parallel efficiencies are shown and the page definitely turned on fast parallel algorithms, their application to this special case will be briefly discussed.

3.4 Symmetry considerations

A symmetry plane, or axis in two dimension, is present in many problems of interest including the application presented in the next chapter. In this situation, only half the problem needs to be solved. An inviscid boundary condition (no-through flow) is enforced along the symmetry axis by the method of image vortices. Assuming that the symmetry axis is the x-axis, the image of a vortex centered at z is located at z^* and is assigned the same circulation as the original vortex but with the opposite sign. The presence of these image vortices makes the x-axis a streamline. The images influence the flow field of the top half but there is no need to evaluate the velocity at their location since they follow the motion of the original vortices. When group A interacts with group B as shown on Fig. (3.4.1), the influence of the group images must be considered as well.

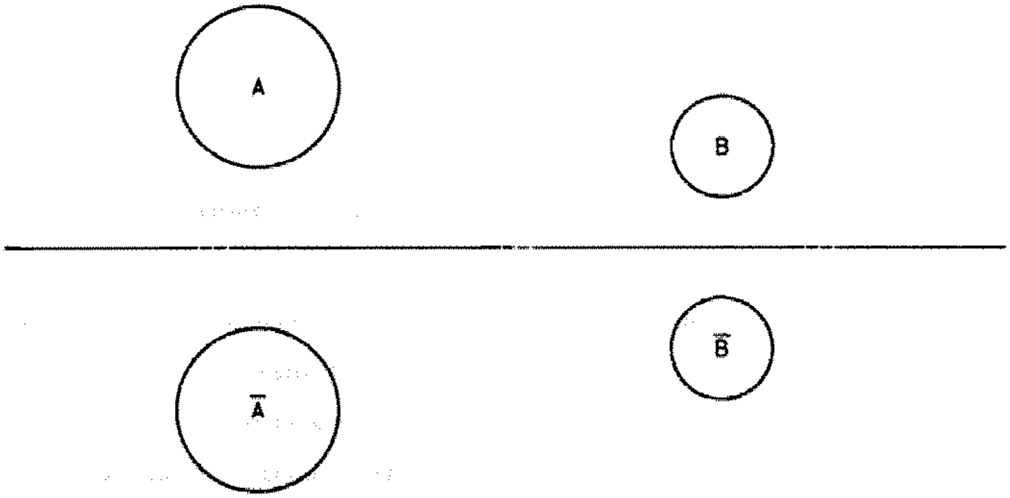


FIGURE 3.4.1 Two groups of vortices and their symmetric images.

The question is: can the influence of Ā on B and B̄ on A be determined from the knowledge of the influence of B and A on each other? The answer is no because $(\bar{z}_A - z_B)$ is not related to $(z_A - z_B)$. However,

$$(\bar{z}_A - z_B) = (z_A - \bar{z}_B)^* \tag{3.4.1}$$

and that symmetry can be exploited. In a first step, every processor makes a copy of its data structure that represents its images. The sign of the circulations and y-locations is changed to capture the reflection process. The multipole coefficients of the reflected data structure do not need to be computed anew since

$$\bar{a}_k = -{a_k}^* \ . \tag{3.4.2}$$

Then, when group A visits the processor responsible for B, it can interact with B and its reflection, $\bar{\mathrm{B}}$. However, the influence of A on $\bar{\mathrm{B}}$ is of no interest since the velocity of the images is not necessary. What is really needed is the influence of $\bar{\mathrm{A}}$ on B and this is where Eq. (3.4.1) becomes useful as it can be used to show that

$$b_k = -\bar{b}_k^* \ , \tag{3.4.3}$$

where the b_k represent the influence of $\bar{\mathrm{A}}$ on B.

When two processors are done interacting with each other, the coefficient $\bar{b}_k$ are added to the b_k following Eq. (3.4.3) and are reset to zero to be ready for the next visitor. The same treatment is applied to the velocities accumulated in the vortices of $\bar{\mathrm{B}}$. Using this little trick, there is no need to reflect the visiting data structure and the **between** can be used to compute the interaction of the visiting vortices with the resident images since the coefficients assigned to the images are not wasted.

3.5 Efficiency of parallel implementation

Since the next chapter will present the simulation of the flow around a cylinder, the efficiency of the parallel implementation was tested on such a problem. A cylinder was surrounded by N vortex blobs and their symmetric images as in Fig. (2.3.1) except that only the region for which $1 < r < 1.6$ was covered with particles. The parallel efficiency, as defined in Eq. (3.3.1), is shown on Fig. (3.5.1) as a function of the hypercube size. The parallel implementation is fairly robust as E remains larger than 0.7 for a 32-node concurrent computer meaning that a typical processor does useful work at least 70% of the time. The number of vortices per processor was kept roughly constant at 1500 even if the parallel efficiency is not a strong function of the problem size. It is, however, much more sensitive to the quality of the domain decomposition. The fast parallel algorithm performs better when all the sub-domains have approximately the same squarish shape or in other words, when the largest group assigned to a processor is as compact as possible.

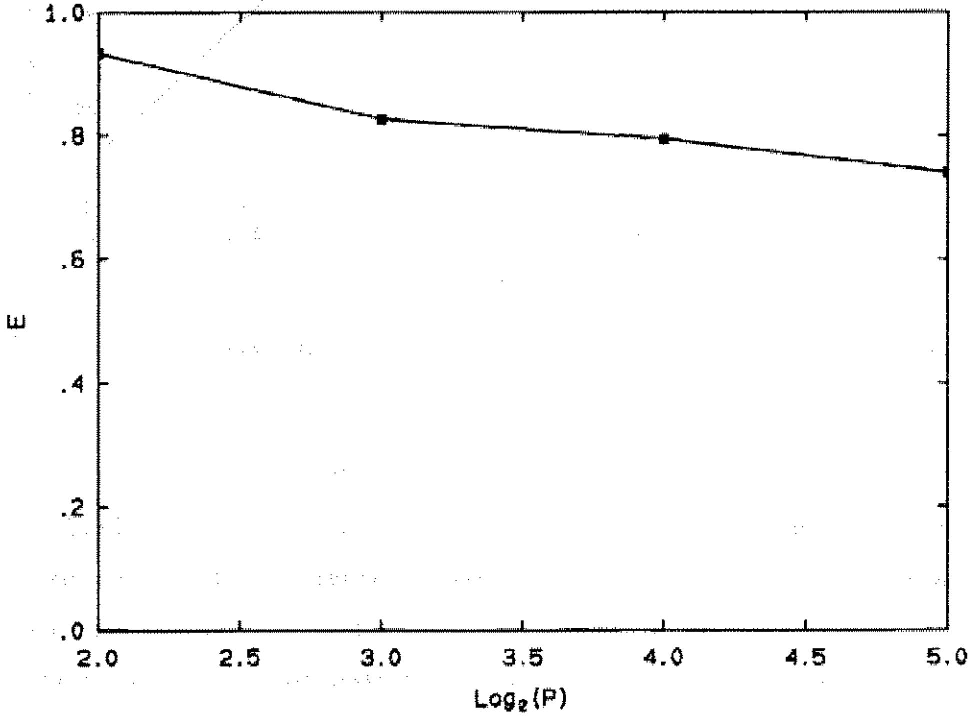


FIGURE 3.5.1 Parallel efficiency of the fast algorithm.

The results of Fig. (3.5.1) were obtained at early times when the Lagrangian particles are still distributed evenly around the cylinder which makes the domain decomposition an easier task. At later times, the distribution of the vortices does not allow the decomposition of the domain into P groups having approximately the same radius and the same number of vortices. Some subdomains cover a larger region of space and as a result, the efficiency drops to approximately 0.6. This is mainly due to the fact that more processors end up in the near-field of a processor responsible for a large group; the request lists are longer and more data has to be moved between processors.

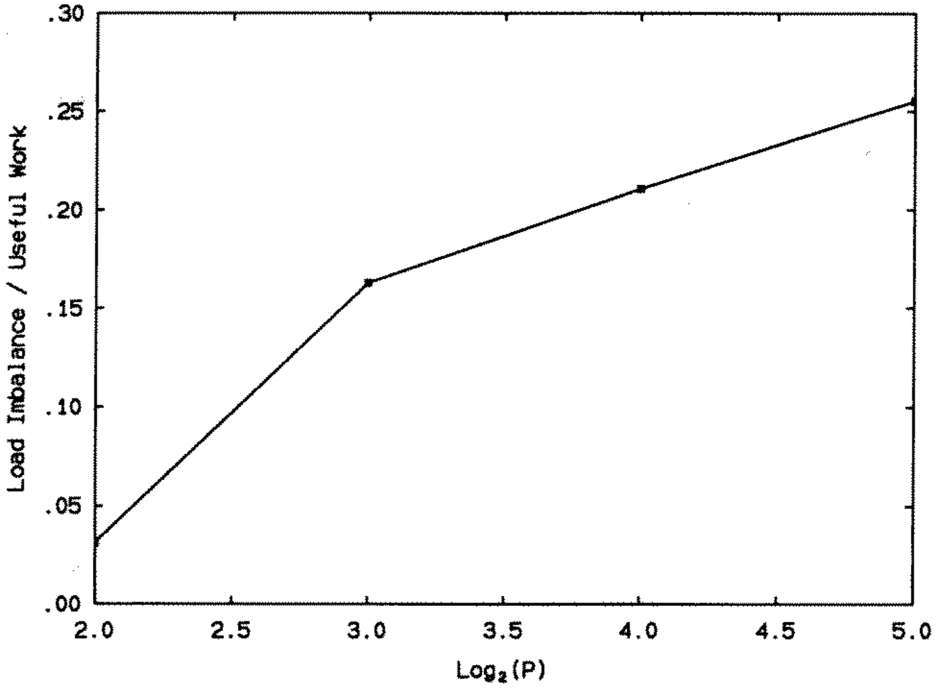


FIGURE 3.5.2 Load imbalance as a function of the number of processors.

The normalized sources of overhead corresponding to Fig. (3.5.1) are shown on Figures (3.5.2), (3.5.3) and (3.5.4). Load imbalance, the largest overhead contributor, is defined as the difference between the maximum useful work reported by a processor and the average useful work per processor. It is a measure of how much faster the simulation would have been if the load had been equally divided among the processors. Secondly, the extra work includes the time spent making a copy of one's own data structure, the time required to absorb the returning information and the work that was duplicated in all processors, namely, the search for acceptable interactions in the upper portion of the tree and the subsequent creation of the request lists. The remaining overhead has been lumped under communication

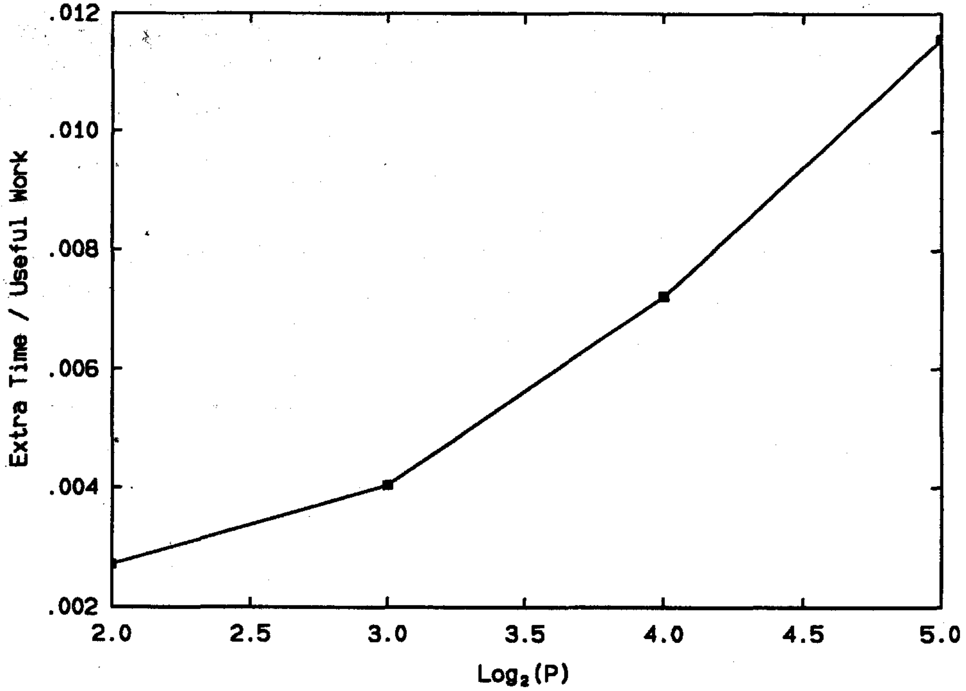


FIGURE 3.5.3 Additional work required by the parallel implementation as a function of the number of processors.

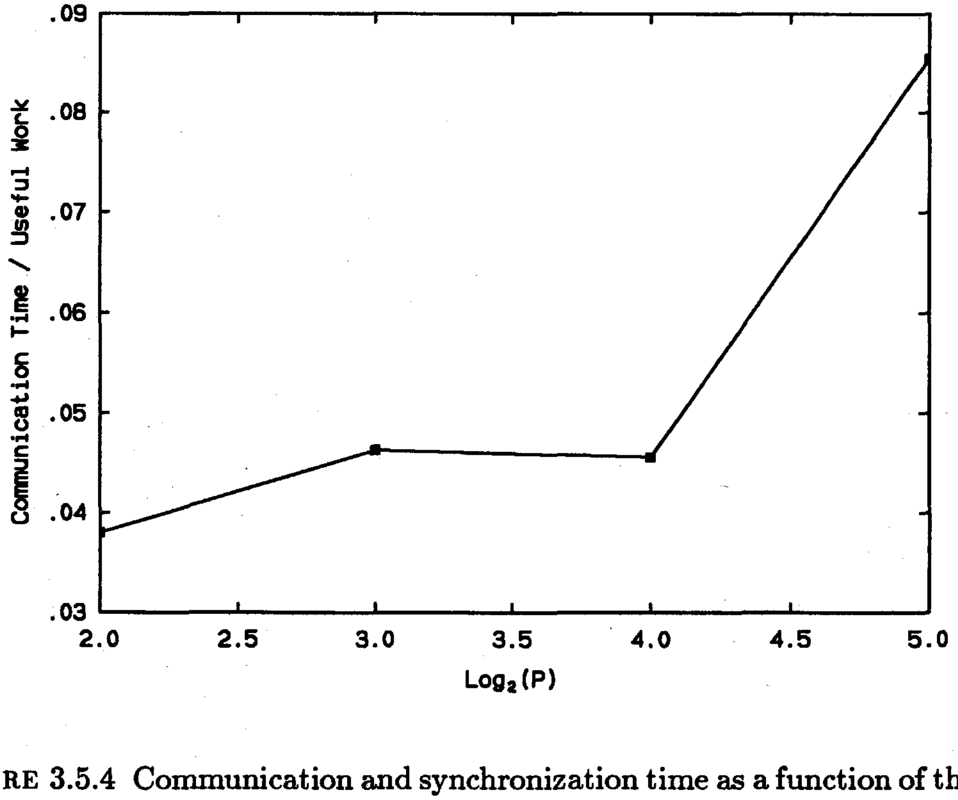


FIGURE 3.5.4 Communication and synchronization time as a function of the number of processors.

time although most of it is probably idle time (or synchronization time) that was not included in the definition of load imbalance. It was originally expected that as P increases, the near-field of a processor would eventually contain a fixed number of neighboring processors. The length of the request lists and the load imbalance would then reach an asymptote and the loss of efficiency would be driven by the much smaller communication and extra times. However, this has yet to happen at 32 processors and the communication time is already starting to make an impact. Nevertheless, the fast algorithm, its reasonably efficient parallel implementation and the speed of the MarkIII have made the large scale simulations of Chap. 4 possible.

CHAPTER 4

Flow past an impulsively started cylinder

This chapter discusses the application of the numerical scheme described in the previous chapter to the simulation of the early stages of the symmetric wake development behind a circular cylinder impulsively brought from rest to a constant velocity U. To enforce the no-slip boundary condition, vorticity is created on the surface of the moving cylinder. This vorticity diffuses from the wall to the fluid where it is convected toward the rear stagnation point of the cylinder. In time, enough vorticity accumulates in this region to form the primary vortices and for $Re > 4.4$, causes flow separation on the back of the cylinder. As the recirculating flow gets stronger, it might itself separate and re-attach to the cylinder surface creating secondary eddies. As observed experimentally by Bouard & Coutanceau (1980), the nature of the secondary phenomenon is very sensitive to the Reynolds number and therefore, should provide a severe test on the ability of the proposed numerical scheme to capture finite Reynolds number effects.

The fast viscous vortex method presented in the preceding chapters is used to compute this flow and the numerical results are compared with the abundant experimental data of Bouard & Coutanceau (op. cit.) available at $Re = 550$, 3000 and 9500. The Reynolds number is based on the cylinder diameter , $Re = \frac{UD}{\nu}$, but its radius is used to normalize the time, t, according to

$$T = \frac{Ut}{R} \ . \tag{4.1}$$

For example, a non-dimensionalized time, $T = 3$, means that the cylinder has been displaced by three radii. Bouard & Coutanceau were able to maintain a symmetric development for six time units at the two lower Reynolds numbers and for four time units at $Re = 9500$. In the present calculation, as in the experiments, the cylinder is moved toward the left and the closed wake appears to its right. The flow features that are used to compare the numerical and experimental data are described in the next section.

4.1 Diagnostics

The ability of the numerical scheme to capture finite Reynolds number effects is mainly assessed by comparing the numerical solution with experimental observations. Streamlines are used to determine the general behavior of the flow field and more specifically, the nature of the secondary phenomena. The time history of drag coefficients (total and form drag) obtained from the simulations are compared with theoretical results valid for short times, since experimental results are not available.

It should be noted that these diagnostics are inherently robust as they are based on integrals of the vorticity field. The streamfunction for example is found by integrating the vorticity twice. The vorticity field does not need to be pointwise accurate to produce streamlines that look satisfactory. The same argument can be applied to the recirculating velocities obtained from the Biot-Savart integral. The evaluation of the drag coefficients also involve an integration of the vorticity field but are not as inherently robust and provide a better assessment of the quality of the numerical solution. These diagnostics are less forgiving because the vorticity integral is differentiated with respect to time in the case of the total drag and heavily biased toward the near-wall region for the form drag.

4.1.1 Streamlines and pathlines

The experimental work of Bouard & Coutanceau is based on flow visualization. The flow field is briefly illuminated with a strong light sheet perpendicular to the cylinder axis and the motion of reflective particles is captured on film. The resulting pathlines provide a good qualitative description of the velocity field. From a computational perspective, it is much easier to describe the flow with streamlines.

Numerically, the contribution of the vortices to the streamfunction is found by solving

$$\nabla^2 \psi_\omega(\mathbf{x}, t) = -\omega(\mathbf{x}, t) \ . \tag{4.1.1}$$

Again, a Green's function technique is used and the streamfunction associated with the regularized vortex particles

$$\omega(\mathbf{x},t)=\sum_j^N \alpha_j(t)\,R_\sigma\left(\mathbf{x}-\mathbf{x}_j(t)\right) \tag{4.1.2}$$

is

$$\psi_\omega(\mathbf{x},t)=\sum_i^N \frac{\alpha_j}{2\pi}\,\chi(|\mathbf{x}-\mathbf{x}_j|/\sigma)\,\log(|\mathbf{x}-\mathbf{x}_j|)\ . \tag{4.1.3}$$

The function $\chi(\rho)$ is a byproduct of the regularization and, for a Gaussian blob,

$$\chi(\rho)=\frac{1}{2}\left(\log(\frac{\rho^2}{2})+E_1(\frac{\rho^2}{2})\right)\ , \tag{4.1.4}$$

where E_1 is the exponential integral. As for the velocity field, a fast algorithm can be used to evaluate the summation. Groups are identified and their multipole expansion coefficients are identical to those used by Greengard & Rokhlin (1987). An approach due to Barnes & Hut (1986) is used to evaluate the influence of the data structure on an Eulerian grid. The contribution from the uniform flow impinging on the cylinder is then added to that of the vortices,

$$\psi(\mathbf{x},t)=\psi_\omega(\mathbf{x},t)+Uy\ . \tag{4.1.5}$$

Streamfunction contours can then be plotted and compared with experimental pathlines. In unsteady flows, pathlines are not necessarily tangent to the velocity field and a close examination of Bouard & Coutanceau's pictures reveals crossing pathlines, an indication that significant unsteadiness occurred while the film was exposed. This is most noticeable at $Re=9500$. Unfortunately, the experimentalists did not provide exposure times over which the streamfunction was in effect integrated to produce pathlines. Nevertheless, it is felt that the pathlines and streamlines can be qualitatively compared as the general features of the flow field should appear in both description.

In addition to the qualitative description of the flow patterns, Fig. (4.1.1) shows geometrical features of the closed wake that can be derived from the knowledge of the streamfunction. The center of the primary vortex is an extremum of ψ and this definition is used to find the coordinates, (a,b) of the primary vortex. The separation angle θ_s is available as well by defining separation as the point where streamlines are no longer parallel to the surface and the streamline $\psi = 0$ breaks away from the wall. Finally, the extent of the wake L is defined as the location where the $\psi = 0$ streamline intersects the symmetry axis. Since these quantities are experimentally derived from pathlines, they are subjected to the same caveat as the flow patterns.

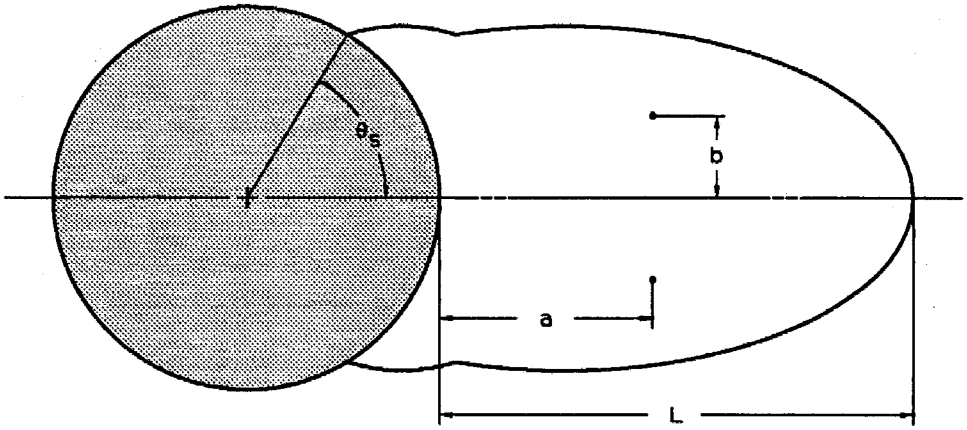


FIGURE 4.1.1 Closed wake geometrical parameters

By measuring the length of the traces left by the reflective particles, Bouard & Coutanceau also determined the velocity on the symmetry axis behind the cylinder. The strength of that recirculating flow is compared to velocities derived from the numerical vorticity field with the help of the Biot-Savart kernel.

4.1.2 Drag coefficients

The transient behavior of the total and form drag coefficients can also be extracted from the numerical solution. Even though they are of considerable engineering interest, experimental results for these quantities are scarce. Schwabe (1935) indirectly determined the form drag on an impulsively started cylinder at $Re = 580$ by manipulating the velocities obtained from flow visualization. More recently, Bingham et al. (1952) and Sarpkaya (1966) directly measured the transient behavior of the total drag on a cylinder. Unfortunately, they did so at Reynolds numbers higher than the ones presented here. However, the computed drag coefficients can still be compared to theoretical results and to other numerical simulations.

Numerically, the total drag per unit span is found by differentiating the fluid linear impulse with respect to time,

$$D = -\frac{d}{dt}\iint_{\Omega} \rho\, u(\mathbf{x},t)\, d\mathbf{x} \ , \tag{4.1.6}$$

which can also be written as

$$D = -\frac{d}{dt}\iint_{\Omega} \rho\, y\, \omega(\mathbf{x},t)\, d\mathbf{x} \tag{4.1.7}$$

in the absence of accelerating boundaries. In the problem of interest, Eq. (4.1.7) is valid as soon as the cylinder has reached its terminal velocity. When the discrete expression of the vorticity field is substituted in Eq. (4.1.7), it is found that

$$D = -\frac{d}{dt}\sum_{i}^{N} \alpha_i\, y_i \ . \tag{4.1.8}$$

The circulation exchange and the motion of the particles can contribute to the drag on the cylinder, as well as the vorticity flux emanating from the viscous panels. The summation in Eq. (4.1.8) is evaluated at the end of the viscous fractional step. At that time, the vorticity created at the wall has already reached the fluid particles and the viscous panels do not directly contribute to the summation.

A non-dimensional drag coefficient is obtained by normalizing D with respect to the force per unit span that would be exerted by the fluid stagnation pressure on the frontal area of the cylinder;

$$C_D = \frac{D}{\rho R U^2} \ . \tag{4.1.9}$$

It is also possible to determine the profile or form drag from the vorticity creation at the walls,

$$C_{D_P} = \frac{D_p}{\rho R U^2} = \frac{2}{\rho U^2} \int_0^{\pi} \nu \left(\frac{\partial \omega}{\partial r} \right)_{\text{wall}} \sin\theta \, d\theta \ , \tag{4.1.10}$$

where the vorticity flux, $\nu \frac{\partial \omega}{\partial r}$ is derived from the observed slip velocity.

Using the no-slip condition, the skin friction coefficient can be expressed as an integral of vorticity at the wall,

$$C_f = \frac{2\nu}{U^2} \int_0^{\pi} \omega(R, \theta) \sin\theta \, d\theta \ . \tag{4.1.11}$$

This integral is difficult to evaluate from a Lagrangian description of the vorticity field since there are no particles and, hence, no accurate information at the wall itself. The regularized vorticity field could be substituted in Eq. (4.1.11), but instead, the skin friction drag is evaluated by subtracting the form drag from the total drag. The cylinder does not experience a lift force as long as the flow remains symmetric with respect to the x-axis. This is always the case in the present numerical simulations.

4.2 Numerical Considerations

Some numerical choices apply to all simulations shown in this chapter.

- Symmetry with respect to the horizontal axis is enforced with image vortices.
- The vorticity field is regularized with Gaussian cores.
- Straight panels are used to represent the surface of the cylinder.
- The vorticity flux is constant for each of these panels.

Before proceeding with the results of the simulations, two important points need to be discussed. First, it will be shown that the no-through flow condition is automatically satisfied when the no-slip condition is enforced. By selecting an appropriate time integration scheme, the inviscid boundary condition can simply be ignored resulting in significant computational savings. Then, the question of remeshing is addressed. The need for this delicate procedure is discussed below, as well as the details on how it is implemented in practice.

4.2.1 Boundary conditions

As mentioned in Chap. **2**, one of the purposes of the viscous fractional step is to enforce the no-slip condition at the wall. Moving the vortex blobs during the convective fractional step results in a slip velocity at the wall. That velocity is canceled during the viscous fractional step with an appropriate vorticity flux determined from

$$\nu \frac{\partial \omega}{\partial r} = \frac{u_s}{\Delta t} \ . \tag{4.2.1}$$

For a cylinder, the vorticity always emanates radially from the wall. The cylinder surface is discretized into a set of M straight viscous panels and the slip velocity is sampled along each of them. Since the vorticity flux within a panel is assumed to be constant, the average slip velocity is sought and substituted in Eq. (4.2.1) to determine the magnitude of the flux. Using a velocity potential formulation, $\mathbf{u} = \nabla \phi$ and the average slip experienced by the panel "i" is

$$
\begin{aligned}
(\bar{u}_s)_i &= \frac{1}{\Delta s} \int_{s_i}^{s_{i+1}} \frac{\partial \phi}{\partial s}\, ds \\
&= \frac{(\phi_{i+1} - \phi_i)}{\Delta s} \ ,
\end{aligned}
\tag{4.2.2}
$$

where ϕ_i and ϕ_{i+1} are the velocity potential evaluated at the panel edges and Δs is the length of the panel. Each vortex blob contributes to the potential according to

$$
\phi_\omega(\mathbf{x}, t) = \frac{-\alpha}{2\pi}\, \chi(|\mathbf{x} - \mathbf{x}_\alpha|/\sigma) \arctan\left(\frac{y - y_\alpha}{x - x_\alpha}\right) \ , \tag{4.2.3}
$$

where $\chi(|\mathbf{x}|/\sigma)$ is the same regularization function as that used for the streamfunction. Finding the average slip consists in determining the influence of the free stream and of the N vortex particles at the M control points that define the panels. In a fast algorithm context, it is crucial not to end up with a slow boundary condition. Just like the streamfunction, the velocity potential due to the vortex particles can be determined using multi-range approximations. As a result, the computational effort needed to enforce the no-slip boundary condition grows like $\mathcal{O}(M \log N)$.

Under special circumstances that will be described below, the no-slip is the only boundary condition that needs to be enforced. Switching back to the streamfunction formulation where ψ includes the contribution of the incoming uniform flow and of the vortex particles, the no-slip condition at the wall is written as

$$
\left.\frac{\partial \psi}{\partial r}\right|_{\text{wall}} = 0 \ . \tag{4.2.4}
$$

This condition is approximately met at the end of the viscous fractional step when the vorticity flux has left the panels and reached the fluid. The enforcement of the no-slip condition coincides with the beginning of the convective fractional step. Convection is subject to the no-through flow condition which can be written as

$$
\left.\frac{\partial \psi}{\partial \theta}\right|_{\text{wall}} = 0 \ , \tag{4.2.5}
$$

where $\theta = 0$ at the rear stagnation point and increases in the counter-clockwise direction. It turns out that Eq. (4.2.5) is automatically satisfied when Eq. (4.2.4) is enforced. The argument is similar to Giesing's (1965) except that no-slip occurs as a result of a smooth vorticity field and not from the presence of vortex sheets. To show the equivalence, Green's theorem is applied to the cylinder's interior and

$$\iint_{\text{body}} (\psi \nabla^2 \psi + \nabla\psi \cdot \nabla\psi)\, dA = \int_{\text{wall}} \psi \frac{\partial \psi}{\partial r} R d\theta \ . \tag{4.2.6}$$

The streamfunction inside the cylinder is harmonic and when Eq. (4.2.4) applies, it is found that

$$\iint_{\text{body}} (\nabla\psi \cdot \nabla\psi)\, dA = 0 \ , \tag{4.2.7}$$

which can only be true if $\nabla\psi$ vanishes everywhere inside the body. Since the fluid within the cylinder is stagnant, there cannot be any through flow and the cylinder surface is indeed a streamline. The same argument can be applied to bodies of arbitrary shape as long as their area does not vanish. There is no through flow at the beginning of the convective step when Eq. (4.2.4) is still valid. At this instant, the motion of the vortices can be computed without enforcing an inviscid boundary condition. However, as soon as the vortices are displaced, a slip velocity appears and the cylinder surface is not necessarily a streamline anymore. When an integration scheme requiring sub-steps is used* to march the vortices in time, image vortices or a distribution of surface singularities are needed to enforce the no-through flow condition after the first substep. However, with an Euler or Adams-Bashforth scheme, velocities are always evaluated in a no-slip configuration and the boundary condition is automatically enforced during the inviscid fractional step; gratis! A second order Adams-Bashforth was used in all the simulations presented in this chapter and the vortex particles were never observed to collide with the cylinder.

---

* e.g. Runge-Kutta

4.2.2 Remeshing

The object of remeshing is to project the known vorticity field from a deformed set of particles onto a set of uniformly spaced particles. More specifically, the problem is to find the particle strengths on the new uniform grid that reproduces the previous vorticity field as well as possible. This delicate operation is made necessary by the distortion of the Lagrangian mesh. The Lagrangian particles that were nicely organized at $T = 0$ rapidly loose their cohesion under the stretching and shearing action of the fluid motion. Eventually, the Lagrangian grid becomes too distorted to represent a smooth vorticity field. This is especially true in the vicinity of stagnation points where the fluid is subjected to a high stretching rate. The stretching pulls the particles away from one another and eventually, the overlap is lost. The weight redistribution method cannot capture the diffusion of vorticity across those gaps. Besides becoming inaccurate, the computation eventually becomes viscously unstable as smaller length scales are generated by the distortion of the fluid. This numerical difficulty is similar to that experienced by Lagrangian finite elements or differences, except that the shearing of the grid is also a problem in these more conventional approaches. The solution is the same, however, as it becomes evident that remeshing is a necessary ingredient of any viscous vortex simulation of bluff body flows.

The projection from the old set to the new one is done with an ad hoc technique and its use is justified a posteriori by showing that the vorticity field is not significantly altered. It was noticed that, although severly deformed, the Lagrangian mesh remained coherent in the sense that pairs of vortices that were neighbors at the beginning of the simulation were still neighbors at later times. This observation provides the basis for the following empirical remeshing scheme.

Using the connectivity of the original grid, it is possible to identify the area of fluid that was assigned to each particle. That area is then superimposed over an undistorted grid and a cookie cutter approach is used to reassign the circulations. If 30% of the deformed area of particle "j" lies over the undeformed area of the new particle "i," 30% of the circulation of "j" is given to "i" and so forth. Although circulation preserving, this simple scheme is slightly diffusive as the averaging of the old circulations within the new cells tends to smear out the existing vorticity gradients. The linear impulse is not identically conserved and can be used as a check on the quality of the remeshing. Typically, remeshing took place at every six or seven time steps and the linear impulse was conserved to at least one part in twenty thousand. When the linear impulse is differentiated to find the total drag, even such a small change can have a noticeable effect and most of the jitters in the drag coefficient are associated with remeshing events. However, it seems that the simulations are not sensitive to the interval between remeshings. Simulations were done with remeshing taking place after a different number of time steps. The flow diagnostics at the end of these simulations were not significantly affected by where and how often the mesh was reorganized.

One positive byproduct of this otherwise unpleasant exercise is that remeshing provides an opportunity for the introduction of new particles in the simulation. Particles must cover all rotational fluid but one cannot predict a priori what portion of the fluid will be vortical at the latter stage of the simulation. Even if it were possible, inert particles would have to be carried through the whole simulation only to become active for the last few time steps; a heavy price to pay. After a reorganization of the Lagrangian mesh, the particles carrying a significant amount of vorticity are surrounded by enough inert particles to accommodate the diffusion of vorticity until the next expected remeshing operation. Adding new particles is an easy task since the Lagrangian mesh that follows a remeshing event is nicely structured. The need to carry the "just in case" particles is eliminated and N is conveniently kept to a minimum.

4.3 Convergence study

The purpose of this study is to assess the effect of the numerical resolution on the quality of the computed solution. The numerical experiments were conducted at $Re = 550$, the easiest and cheapest case presented in this chapter. As the Reynolds number is increased, the scales, both is space and time, get smaller and it is not possible to resolve them as well as for the lowest Reynolds number. By identifying the symptoms of an under-resolved simulation, the convergence study also helps to build confidence in the ability of the numerical scheme to capture the physics with limited computational resources.

It would be too expensive to independently explore the influence of the spatial resolution and the time step. Consequently, they are both varied at the same time as shown in Tab. (4.3.1). The three levels of discretization are referred to as coarse, medium and fine. Going from one level to the next increases the required computational effort by a factor of two (approximately). The numerical parameters were chosen in such a way that all factors contribute to a better solution. The finer spatial discretization allows a better representation of the vorticity gradient, a smaller time step reduces integration and splitting errors and, finally, the time step is closer to the optimum viscous time step (see Sec. (2.2.2)) as the resolution is improved. The number of viscous panels representing the cylinder surface is increased as well.

Resolution	σ	σ/h	No. of panels	Δt	$\Delta t/\Delta t_{\text{opt}}$
Coarse	0.0269	1.2	320	0.038	0.382
Medium	0.0199	1.2	432	0.034	0.624
Fine	0.0149	1.2	576	0.030	0.983

TABLE 4.3.1 Numerical parameters of the convergence study.

Fig. (4.3.1) shows the instantaneous streamlines toward the end of the simulations. Contours of the streamfunction are plotted at regular increments of 0.05 between $\psi = -0.5$ and $\psi = 0.5$. Negative contours are plotted with solid lines while dashes are used for positive values. The streamline $\psi = 0$ divides the closed wake from the outer flow and is bolder than the other streamlines to draw attention to its special nature. In order to discern more details near the dividing streamline, twelve additional contours are shown at regular intervals from $\psi = -0.03$ to $\psi = 0.03$. This convention is used throughout Chap. 4 and App. **B**.

Comparing the streamlines of Fig. (4.3.1), it is reassuring to observe that a reduced discretization does not significantly alter the gross features of the computed flow. As the resolution is increased, there is less numerical smearing and the vorticity layer can roll-up more tightly, producing a slightly stronger primary vortex. The secondary eddy is present in all three cases although not quite as well developed for the coarse resolution. It should also be noted that the results obtained with the medium and the fine resolution are nearly identical. This is an indication that the truncation errors are already small at that point and that increasing the resolution

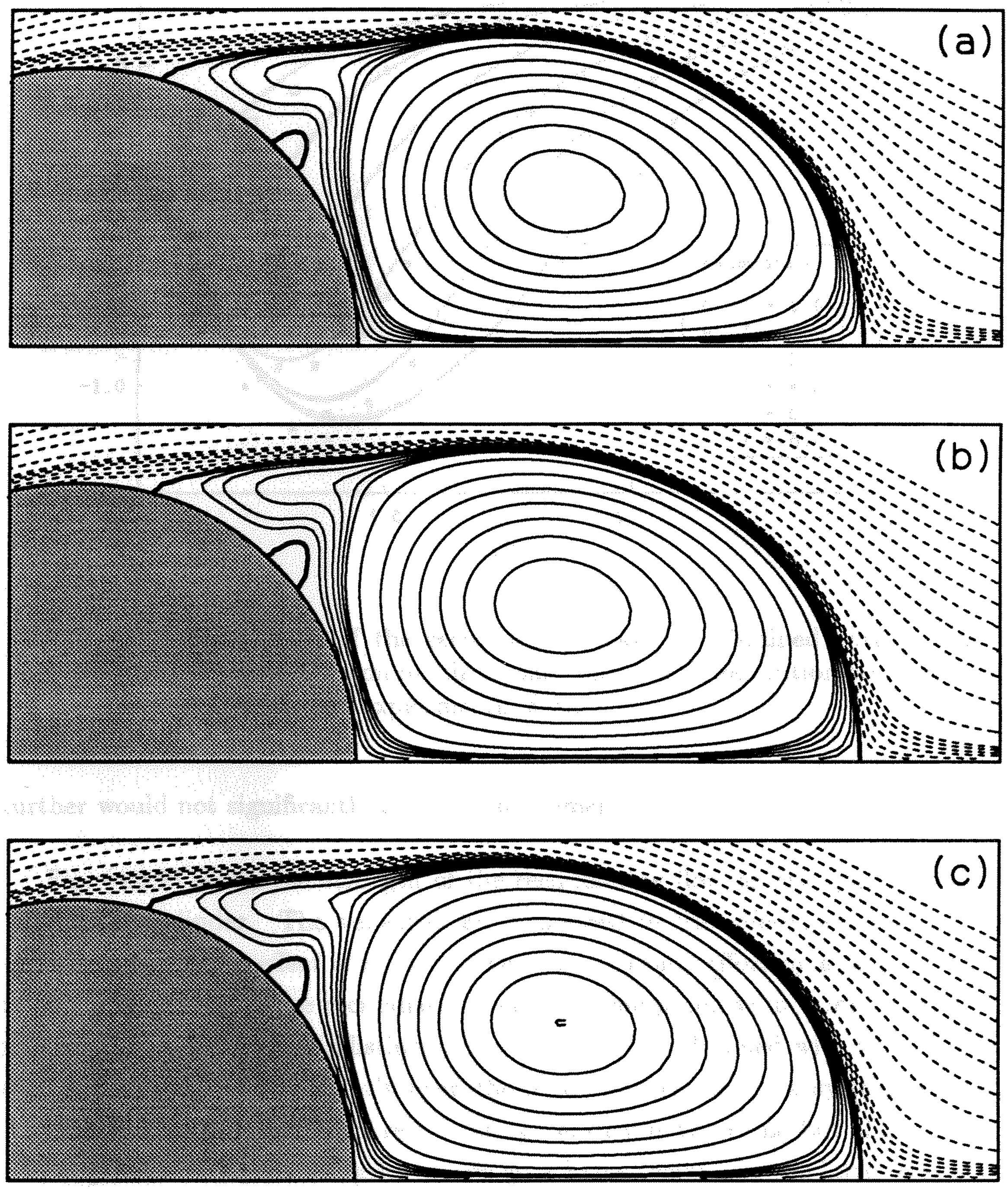


FIGURE 4.3.1 Comparison of the streamlines at $T = 5.5$ for a (a) coarse, (b) medium, and (c) fine resolution.

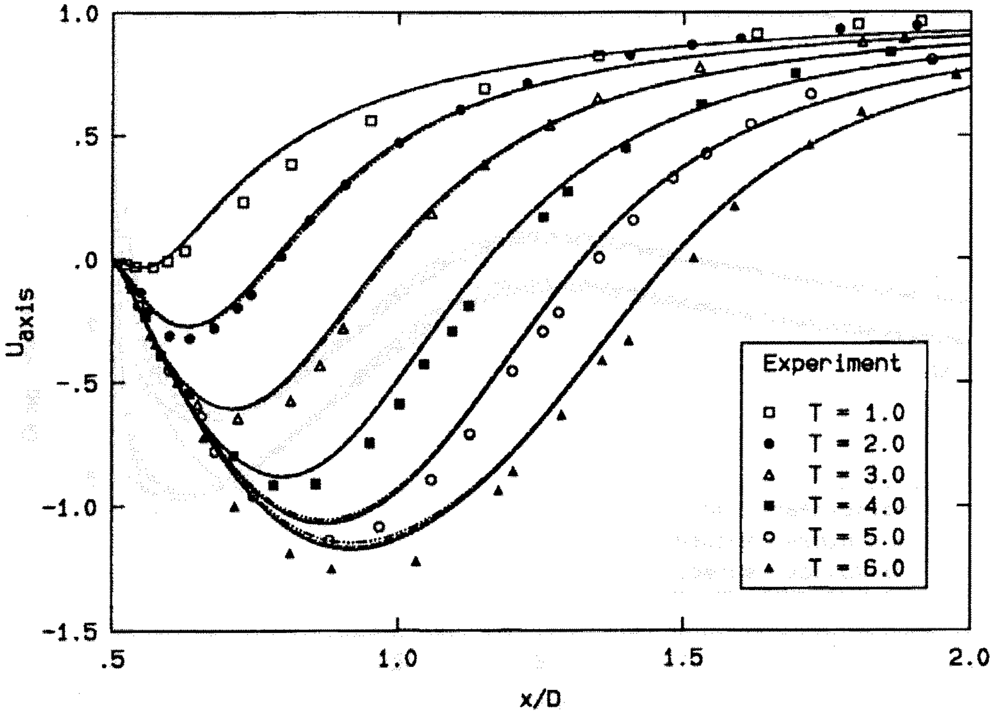


FIGURE 4.3.2 Comparison of the recirculating velocities obtained with a coarse (dot-dash), medium (dash) and fine (solid) resolution with Bouard & Coutanceau experimental data.

further would not significantly improve the numerical solution.

Fig. (4.3.2) shows the strength of the recirculating flow on the symmetry axis behind the cylinder. The velocities are expressed in the reference frame of the cylinder. Again, the effect of reducing the computational effort by a factor of four is barely noticeable. The recirculating velocities computed with a medium and a fine resolution are nearly indistinguishable while those obtained with the coarsest discretization differ only slightly from the other two but not in a systematic way. At early times, the maximum recirculating velocity is the same but the smeared vorticity field of the coarse "grid" produces a longer wake. At the latter stages of the simulation, the length of the wake is approximately the same but deeper velocity wells are observed for the finer discretizations.

More significant discrepancies can be observed in the time history of the drag coefficients. In Fig. (4.3.3), the numerical values are compared with Bar-Lev & Yang's (1975) analytical results. They used matched asymptotic expansions to

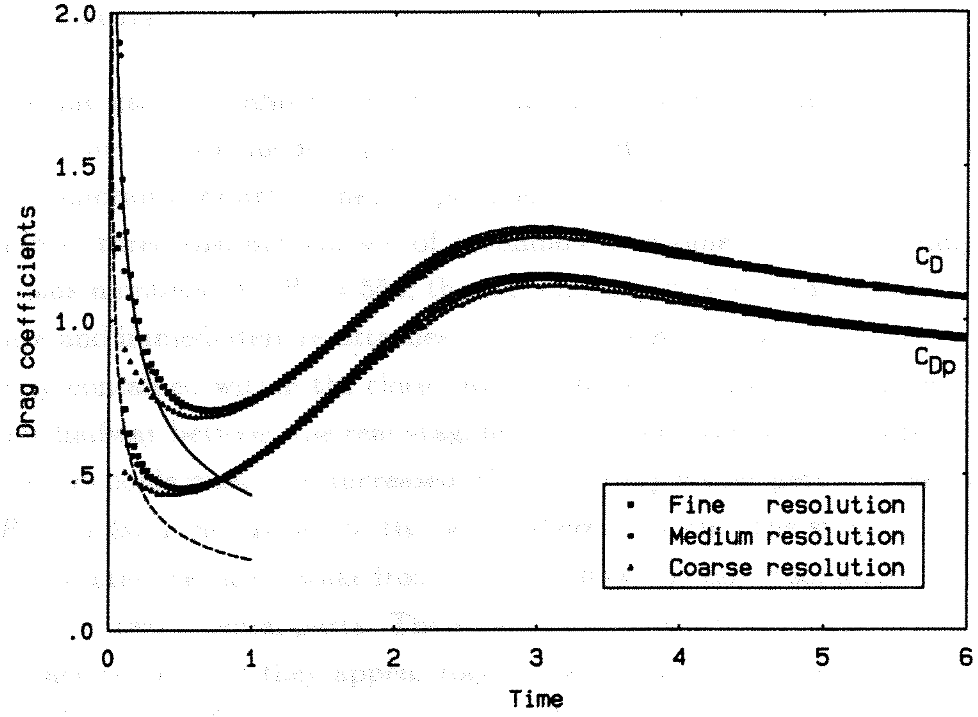


FIGURE 4.3.3 Convergence study of the drag coefficients time history and comparison with Bar-Lev & Yang analytical results (solid and dash lines).

predict the evolution of the vorticity field for short times. Although the rise in the profile drag associated with the formation of the primary eddies is not predicted, their analysis certainly provides an accurate solution for short times. The steep vorticity gradient created by the impulsive motion results in a $1/\sqrt{T}$ singularity near $T = 0$ for both the total and profile drag. Numerically, it is a challenge to capture this transient behavior correctly. The numerical scheme cannot resolve scales smaller than the smoothing length, σ. Steep gradients are smeared over that length and, as a result, vorticity is artificially transported away from the wall. It is seen from Fig. (4.3.3) that the fine resolution hugs the analytical curve more closely. Drag coefficients that drop more abruptly than the Bar-Lev & Yang (op. cit.) solution should be interpreted as a symptom of an under-resolved calculation. The coefficients eventually recover but the effects of the numerical smearing are not confined to short times. Fig. (4.3.3) shows that the less tightly bounded eddy of the coarse resolution produces a weaker suction peak around $T = 3$. The discrepancy gets smaller as the primary vortex drifts away from the cylinder and it appears that the coarse simulation eventually recovers from the errors made at early times.

4.4 Results

Having gained confidence that the truncation errors are under control, the proposed numerical scheme is applied to the three Reynolds numbers at which Bouard & Coutanceau conducted their experiments. From their flow visualizations, they identified three distinct classes of secondary phenomena corresponding to these Reynolds numbers. At $Re = 550$, the recirculating flow separates from the cylinder surface and immediately re-attaches creating a counter-rotating recirculating region entirely contained within the closed wake. This secondary eddy is located approximately halfway between the rear stagnation point and the primary separation point. As the Reynolds number is increased, the secondary vortex gets stronger and larger. At $Re = 3000$, it has grown to the point where it touches the stagnation streamline that separates the closed wake from the outer flow. By doing so, it splits the primary vortex into two unequal parts. The smaller has approximately the same size as the secondary vortex and they appear together as a pair of counter-rotating secondary eddies. Bouard & Coutanceau referred to this process as the α-phenomenon.

When the Reynolds number is further increased to 9500, the wake exhibits an even richer spectrum of behaviors. Most of the extra activity occurs immediately following the primary separation when the recirculating fluid is still confined to a narrow region behind the cylinder, the "forewake." The smoothly rotating fluid that characterized this stage of the wake development at lower Reynolds numbers is replaced by a coherent vortex appearing roughly in the middle of the forewake. This rapidly rotating structure splits the wake in three distinct regions:

- The portion of the wake dominated by the main vortex itself.
- The area comprised between this vortex and the symmetry axis. This region seems to be loosely connected to its powerful neighbor but does not rotate as fast.
- A region where a pair of small counter-rotating eddies stand between the primary separation point and the main vortical structure.

This peculiar configuration, the so-called β-phenomenon, was also observed at $Re = 5000$. Eventually, the vortex moves away from the cylinder and forms the main wake; a secondary vortex appears on the cylinder surface and the α-phenomenon is observed at both $Re = 5000$ an $Re = 9500$.

4.4.1 Re = 550

This section presents in greater details the numerical results obtained with the highest resolution in the convergence study. The upper half of the cylinder is initially surrounded by 6,500 vortex blobs. As more fluid becomes rotational, new particles are introduced at every remeshing event and, when the calculation is stopped at $T = 6$, the vorticity field is represented by almost 50,000 particles. The cylinder itself is represented by 576 viscous panels where vorticity is created in such a way that the no-slip condition is enforced, on the average, along each panel. This number, 576, also corresponds to the number of particles in immediate contact with the wall. The other relevant numerical parameters are given in Tab. (4.4.1).

$N_{\max}$	σ	σ/h	No. of panels	Δt	$\Delta t/\Delta t_{\text{opt}}$
46735	0.0149	1.2	576	0.030	0.983

TABLE 4.4.1 Numerical parameters for the simulation at $Re = 550$.

A complete time history of the wake development is available in App. B. The isolated secondary eddy, characteristic of the flow at this Reynolds number, is first observed at $T \simeq 2.97$. As expected, it is located halfway between the primary separation and the rear stagnation point. It steadily grows until $T \simeq 3.75$, then stays stable for approximately one time unit and finally starts to shrink as the primary vortex drifts away from the cylinder. However, it is still clearly visible at the end of the simulation when the cylinder has been displaced by six radii. Fig. (4.1.1) compares Bouard & Coutanceau flow visualization with the computed streamlines, both at $T = 5$. The agreement is very good. Not only is the size and location of the secondary vortex accurately predicted, but the details of the flow around it are also correctly described. Even the slight kink on the separation streamline is discernable on both descriptions.

The time evolution of the geometrical parameters of the wake is shown on Fig. (4.4.2) and compared with experimental measurements. The large symbols are used for Bouard & Coutanceau's data while the computed values are plotted at every time step with the corresponding smaller symbols. The simulation accurately tracks the trajectory of the primary vortex but systematically under-predicts the length

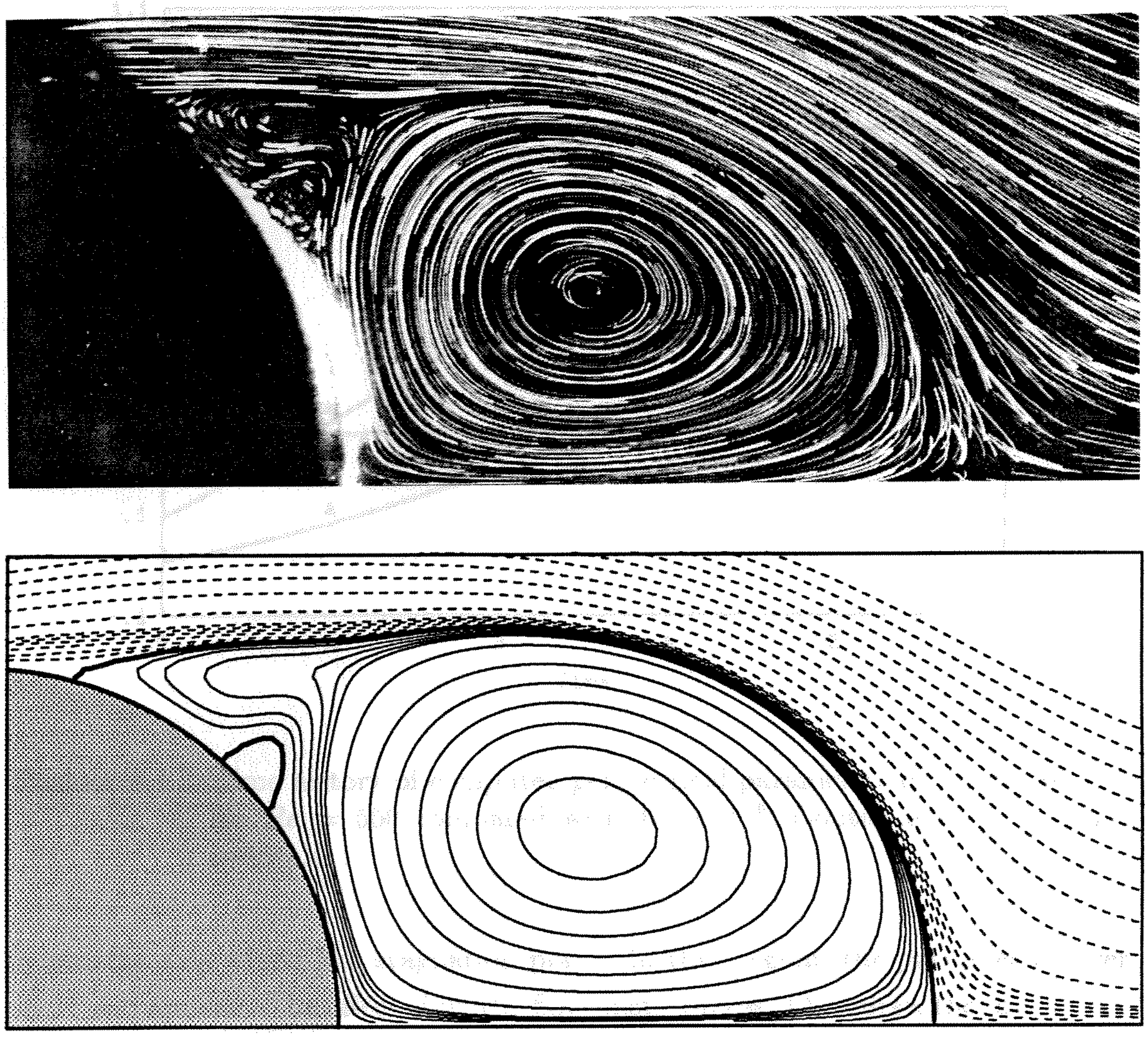


FIGURE 4.4.1 Comparison of computed streamlines with Bouard & Coutanceau experimental flow visualization at $Re = 550$ and $T = 5.0$.

of the closed wake. This a rather surprising results since the additional diffusion introduced by the numerical scheme should produces a fatter primary vortex and hence, a longer wake. Extremely viscous fluids were used in the experiment and the actual Reynolds number could have been slightly lower than 550. Unfortunately, the experimentalists did not provide an error estimate for that quantity or for any other quantity for that matter. Part of the discrepancy could also be attributed to the difficulty of experimentally measuring L from flow visualizations. The extent of the closed wake is defined by the intersection of the separating streamline with the

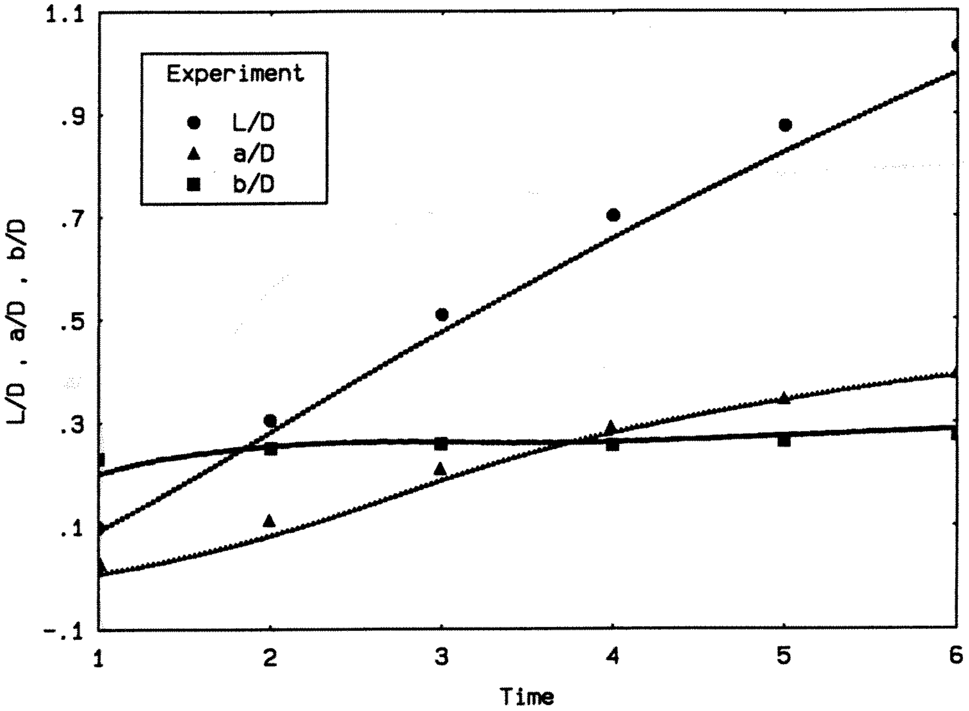


FIGURE 4.4.2 Time history of computed geometrical parameters of the closed wake for $Re = 550$ compared with Bouard & Coutanceau experimental results.

symmetry axis which is a stagnation point. In that region, the pathlines are very short and appear to be affected by the flow unsteadiness. Of course, the discrepancy could also be the expression of numerical truncation errors, especially with respect to the time step, a parameter that the convergence study did not explore over a wide range.

As defined in Fig. (4.1.1), the separation angle is plotted as a function of time on Fig. (4.4.3). Following the primary separation occurring around $T = 0.54$, the recirculating fluid rapidly climbs along the cylinder shoulder. Its progression is progressively slowed down and very little movement of the separation angle can be observed after the formation of the secondary eddy at $T \simeq 3$. This process appears to be smooth and most of the small wiggles are caused by remeshing.

Fig. (4.4.4) shows the drag coefficients as a function of time. The analytical results of Bar-Lev & Yang are also available for comparison. They are plotted up to $T = 1$ which is the approximate range of validity claimed by these investigators.

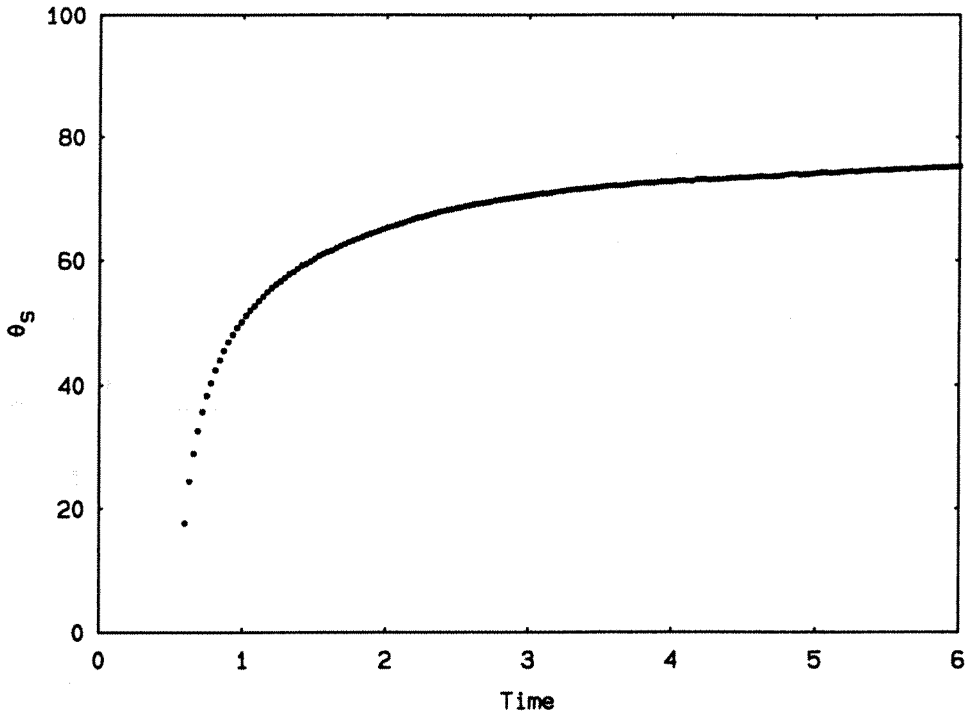


FIGURE 4.4.3 Time history of separation angle for $Re = 550$.

After hugging the asymptotic behavior for a few time steps, the numerical values rapidly depart from Bar-Lev & Yang's prediction. At $T = 0.5$, the vorticity layer is already quite thick and the matched asymptotic expansions are not robust enough to properly represent the vorticity field and the drag that it causes on the cylinder.

Following an abrupt drop from infinity, C_D rises to a maximum of 1.3 at $T \simeq 3$. Most of this increase can be attributed to the profile drag, the skin friction drag staying approximately constant after $T = 1$. The increase in the profile drag is due to the low pressure on the rear face of the cylinder caused by the presence of the primary vortices. On Fig. (4.4.4), the solid triangles represent the time history of the total drag theoretically obtained by Bryson (1959). In his model problem, the vorticity is symmetrically shed from a fixed separation point and is directly lumped into the primary vortex. The strength of the vortex (and its images) is such that the prescribed separation point is also a stagnation point. Bryson (op. cit.) numerically computed the trajectory of the primary eddy and its effects on the pressure drag experienced by the cylinder. Being essentially inviscid, his model is presumably more accurate at high Reynolds number when the vortex layers are thin.

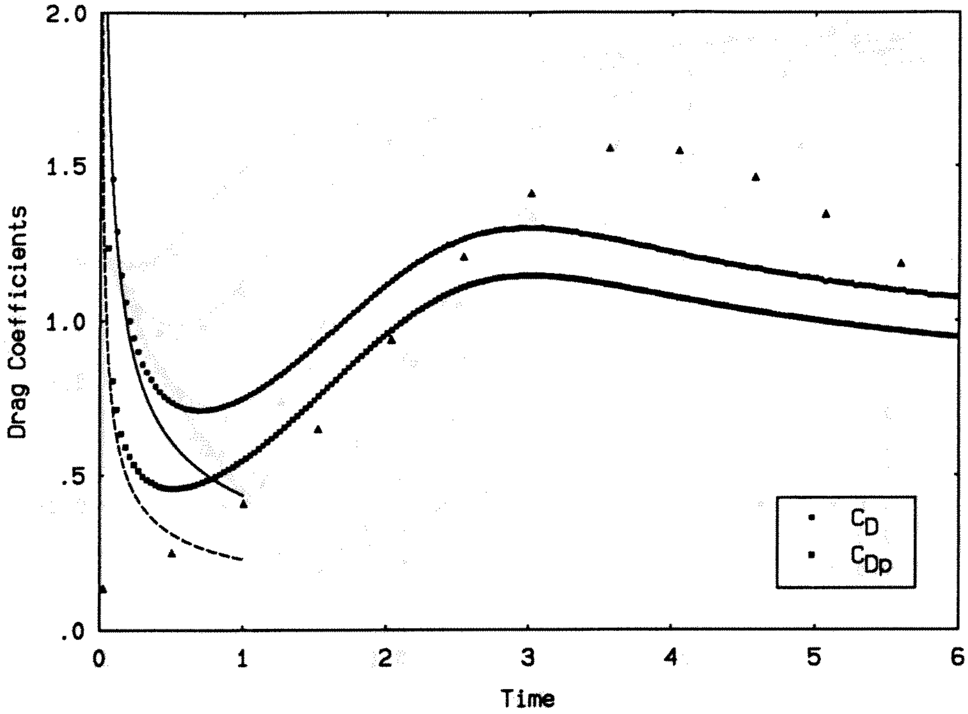


FIGURE 4.4.4 Time history of the computed drag coefficients for $Re = 550$ compared with Bar-Lev & Yang matched asymptotics (solid and dash lines) and Bryson model (triangles).

Considering the simplicity of Bryson approximations, the agreement is remarkable. The key assumption of a separating vortex sheet progressively rolling-up into a vortical structure must be realistic. Indeed, the computed vorticity field shows that the separating vortex layer is smoothly fed into the core of the primary vortex. However, the computed suction peak is not as pronounced as the one predicted by Bryson's model. This could be attributed to viscous effects, absent in the idealized problem. The numerical primary vortex is not only more diffused than Bryson's but it is also weaker as some of its vorticity gets canceled by the vorticity of the opposite sign created on the portion of the cylinder's wall exposed to the recirculating flow. The symmetry axis also acts as a sink of vorticity.

Finally, Fig. (4.4.5) shows the velocity along the "x"-axis in the region immediately behind the cylinder. Owing to the symmetry of the problem, there is no "y"-component of the velocity along this axis. A strong recirculating flow can be observed as the primary eddies induce velocities whose magnitude is larger than that of the incoming flow for $T > 5$. Compared with the experimental results, the

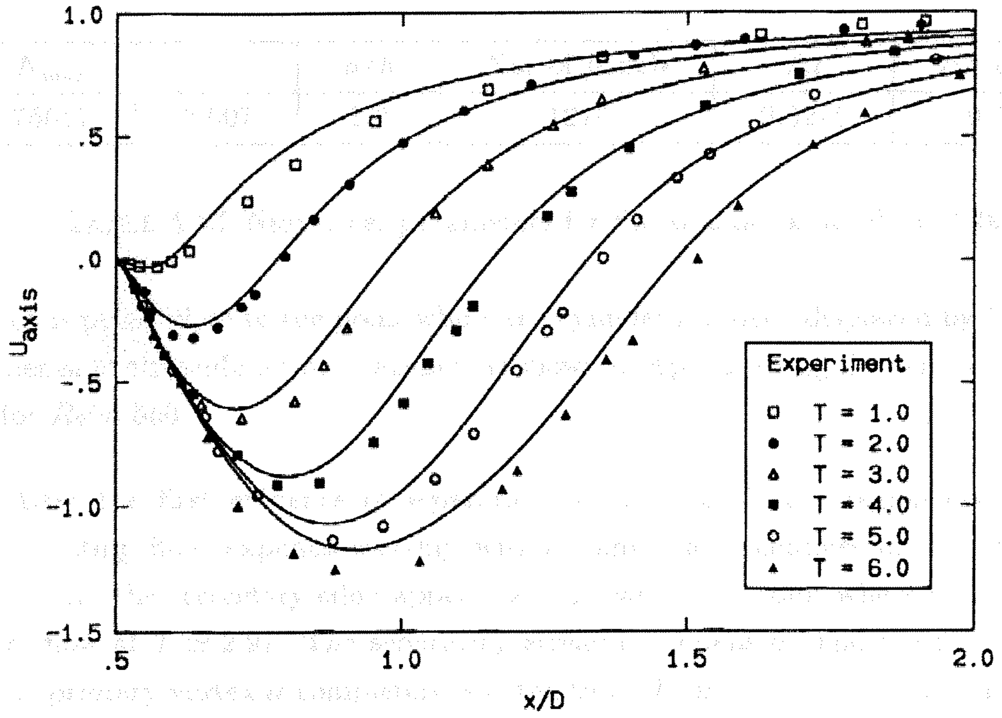


FIGURE 4.4.5 Comparison of the velocity on the symmetry axis for $Re = 550$ obtained numerically (solid lines) and experimentally.

simulation systematically under-predicts the strength of the recirculating flow. This trend is also observed in the velocities computed by Ta Phuoc Loc (1980) using a fourth order compact finite difference scheme. A similar under-prediction was observed for the extent of the closed wake, L. Since that length is defined as the point where $U_{\text{axis}} = 0$, the possible sources of error that were identified for L for both the experiment and the calculation apply for the recirculating velocities as well.

4.4.2 Re = 3000

Steeper vorticity gradients are produced at $Re = 3000$ than at 550 and the spatial resolution has to be significantly increased from the previous case. Mercifully, the wake is more compact and a smaller area of fluid needs to be covered with particles. The numerical parameters selected for the simulation at this Reynolds number are given in Tab. (4.4.2).

Approximately 13,000 vortex blobs are used in the early stages of the simulation

$N_{\max}$	σ	σ/h	No. of panels	Δt	$\Delta t/\Delta t_{\rm opt}$
76017	0.007	1.2	1216	0.0275	0.744

TABLE 4.4.2 Numerical parameters for the simulation at $Re = 3000$.

which is pursued up to the point where the cylinder has been displaced by 5 radii. A sequence of streamfunction snapshots is shown in App. **B** using the same convention as for $Re = 550$.

After the first evidence of separation, at $T \simeq 0.52$, the region occupied by recirculating fluid expands steadily without any major changes until $T \simeq 2$. At this point, the secondary eddy appears and grows to the point where it touches the outer flow at $T \simeq 2.91$. The separating streamline opens up and the front portion of the primary vortex is completely isolated from the main recirculating region until $T \simeq 3.27$. During that time, the secondary eddy interacts directly with the outer flow as fluid (and its vorticity) that has just separated from the cylinder is drawn into the secondary vortex. This phenomenon was not observed in the simulations of Ta Phuoc Loc & Bouard (1985) at $Re = 3000$** and presumably, did not occur in the experiment either. The secondary eddy continues to lose strength until $T \simeq 3.8$ and then enters a second growth phase that lasts up to $T = 5$ when the simulation is stopped.

A tertiary phenomenon, within the secondary eddy, can even be observed after $T \simeq 3.4$. These small scale structures, as well as a second secondary eddy, are also clearly visible in the simulations of Ta Phuoc Loc & Bouard (op. cit.), Lecointe & Piquet (1985) and in the experiments of Nagata, Nagase & Ito (1989).

Fig. (4.4.6) shows a comparison of the computed streamlines with Bouard & Coutanceau experiment at $T = 5$. Again, the agreement is good although the computed wake appears to be stockier than its experimental counterpart. This can be partly attributed to the difficulty of identifying the separation pathline. Nevertheless, the shape and location of the secondary eddy is accurately predicted, as well as the presence of the tertiary eddy and the bulging of the streamlines within

---

** It was observed at $Re = 9500$ however.

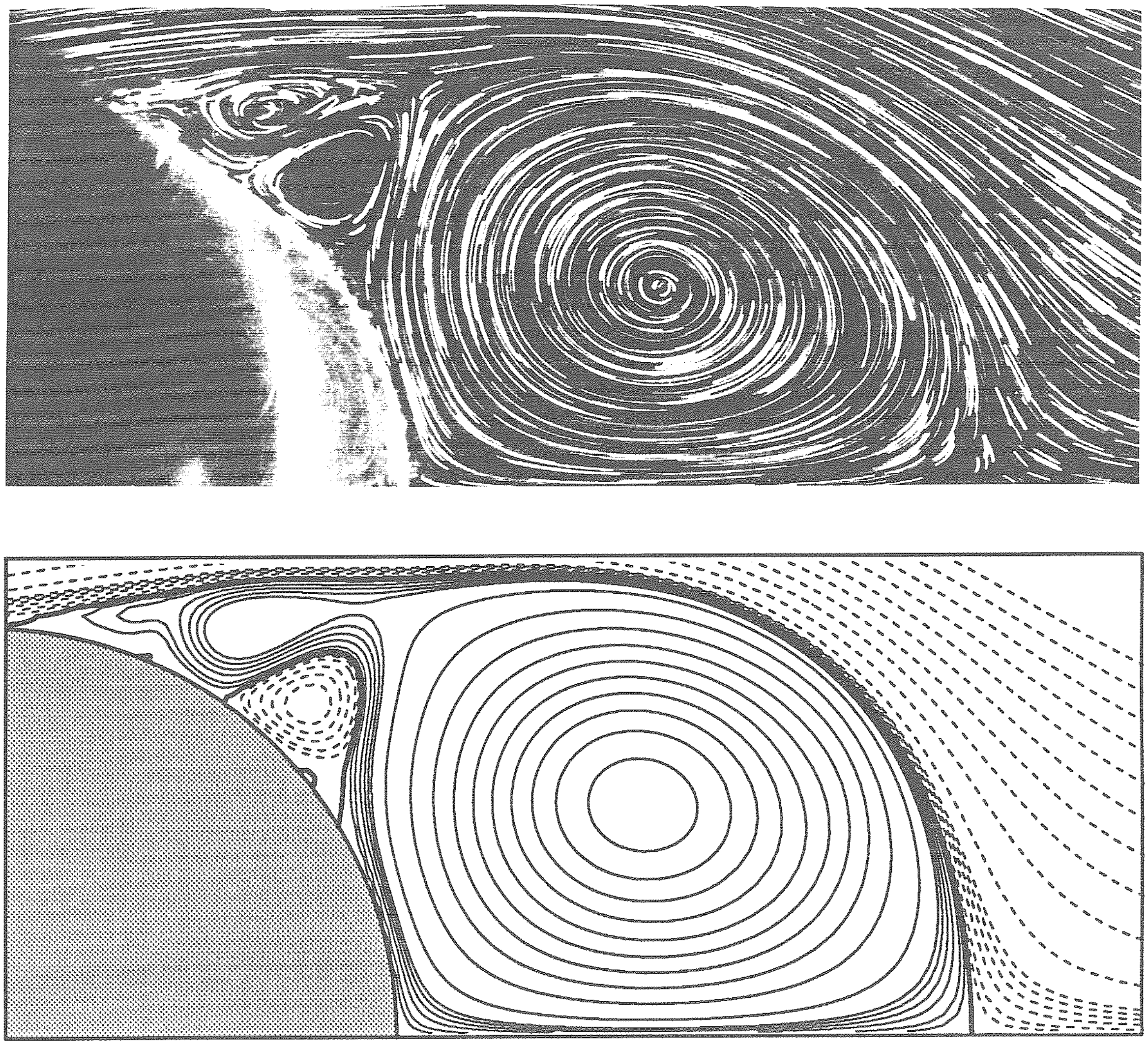


FIGURE 4.4.6 Comparison of computed streamlines with Bouard & Coutanceau experimental flow visualization at $Re = 3000$ and $T = 5.0$.

the recirculating region next to the primary separation.

The geometrical characteristics of the wake are given on Fig. (4.4.7) as a function of time. Immediately following the primary separation, the recirculating flow is weak, making the identification of its center a difficult task, both numerically and experimentally. As it gets stronger, it becomes easier to pinpoint the core of the primary eddy and its trajectory is accurately predicted by the numerical simulation. The largest discrepancies are observed at $T = 3$ where the simulation seems to

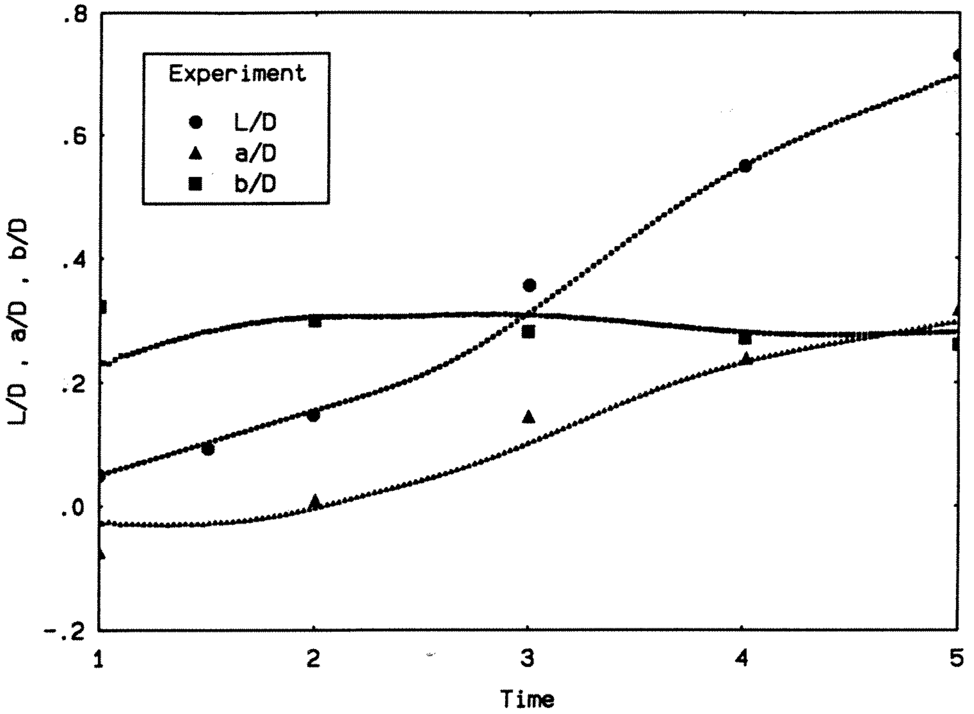


FIGURE 4.4.7 Comparison of the time history of the computed geometrical parameters of the closed wake for $Re = 3000$ with Bouard & Coutanceau measurements.

momentarily depart from Bouard & Coutanceau's observations.

Fig. (4.4.8) shows the progression of the separation angle along the rear portion of the cylinder. The process is similar to what was observed at $Re = 550$ except that separation occurs a little bit sooner, $T \simeq 0.52$ instead of $T \simeq 0.54$. When the simulation is stopped at $T = 5$, the separating streamline leaves the cylinder at approximately 86° which is almost 10° more than at the lower Reynolds number. Fig. (4.4.8) also reveals a small dip in the progression of the separation angle at $T \simeq 3$. Clearly, the smooth evolution of the flow is disrupted around that time and nowhere is it more evident than on the time history of the drag coefficients shown on Fig (4.4.9).

The most striking feature observed on Fig. (4.4.9) is the double hump configuration of the profile drag. At $Re = 3000$, Bryson's dromadery has been replaced by a camel. Clearly, his model does not apply anymore as the progressive rolling up of a separating vortex sheet into a drifting vortex core cannot provide a mechanism

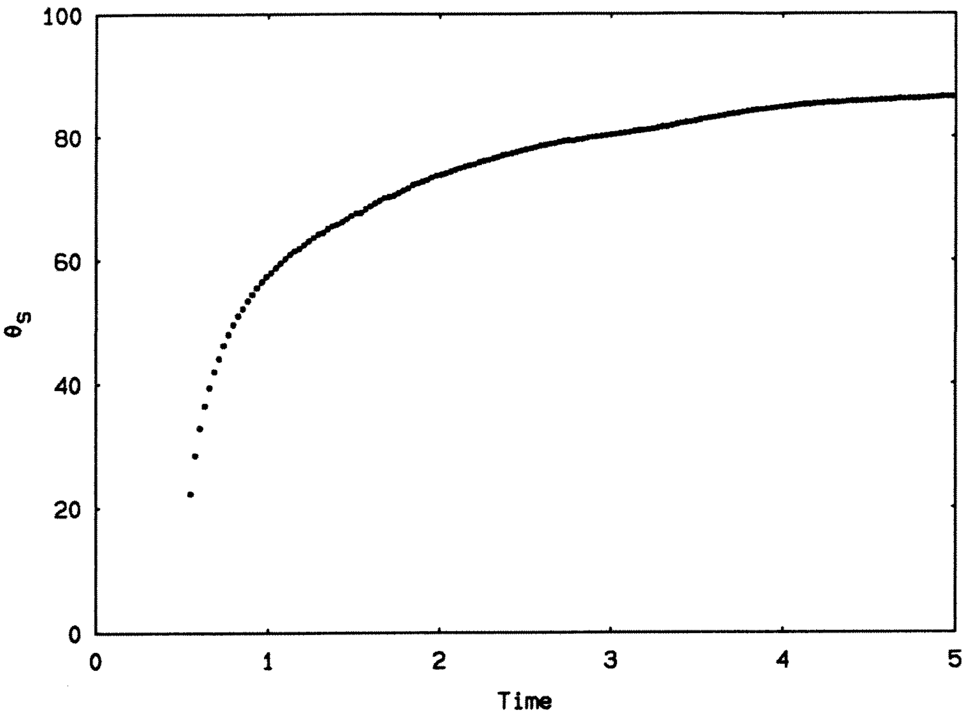


FIGURE 4.4.8 Time history of separation angle for $Re = 3000$.

that would explain the roller coaster of Fig. (4.4.9).

A close examination of the vorticity field reveals that the secondary eddy interferes with the progression of the separating vorticity layer toward the primary vortex. When the secondary eddy has grown to the point where it bursts into the outer flow, the vorticity that had just left the body is drawn back toward the cylinder. The feeding of the primary vortex is interrupted and a new vortical structure begins to roll-up. This structure does not produce drag as effectively as the primary vortex and the drag curves momentarily level off. This structure creates an intense suction peak but, due to its location near the top of the cylinder, its contribution to the drag is minimal. Its presence also weakens the secondary eddy and, eventually, the separating streamline is such that the flux of vorticity can reach the primary vortex again. Most of the vorticity temporarily accumulated in the new structure is carried downstream and the pressure drag starts to rise again. By that time, the primary vortex has drifted away from the cylinder and cannot induce a secondary eddy that would be strong enough to redirect the flow of vorticity. The process leading to the isolation of the primary vortex is not repeated again.

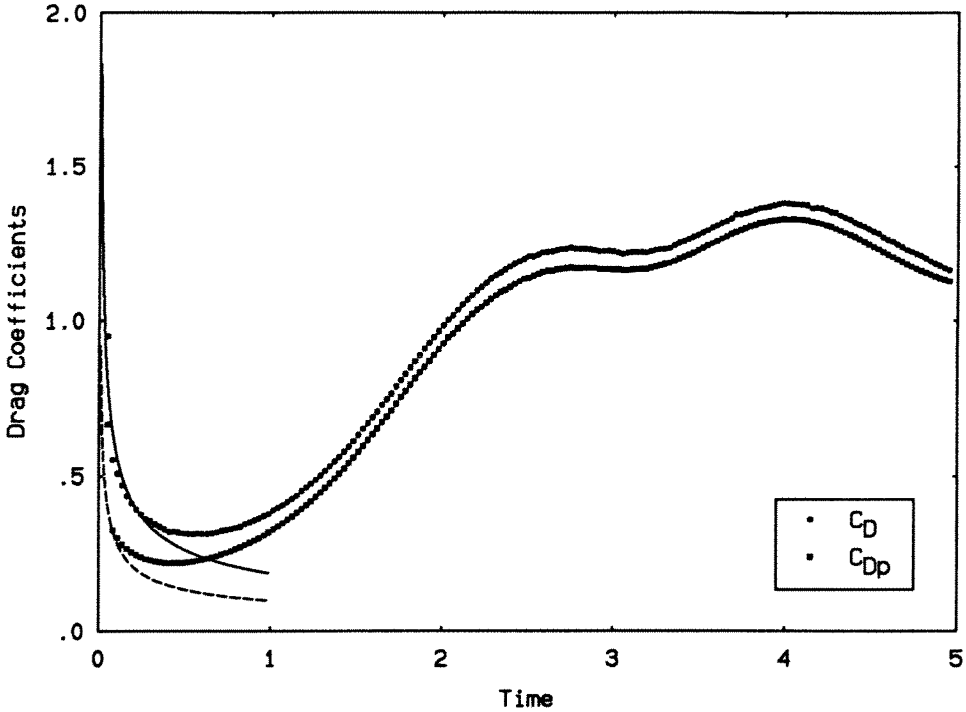


FIGURE 4.4.9 Comparison of time history of the numerical drag coefficients with Bar-Lev & Yang analytical solution (solid and dash lines) for $Re =$ 3000.

This major flow reorganization was not observed in the finite difference simulation of Ta Phuoc Loc & Bouard nor in Bouard & Coutanceau's experiment. Nevertheless, a case can be made that this peculiar behavior is more than a numerical artifact. From the experimental perspective, Nagata and his collaborators used electrolytic precipitation to visualize the motion of the separating fluid. At $Re = 3000$, their streaklines clearly show that the vortex sheet that feeds the primary vortex has been severed and is rolling-up between the primary separation and the secondary eddy. Another flow visualization obtained by Nagata at $Re = 2000$ and published in Nakayama's (1988) compilation of flow visualizations (Fig. 15), dramatically shows the pinching off of the feeding vortex sheet. On the numerical side, Smith & Stansby (1988) used a random vortex method at $Re = 1000$ and obtained a total drag coefficient exhibiting a behavior similar to the one shown on Fig. (4.4.9). Unfortunately, most investigators do not report their drag coefficients and it is not possible to determine if the double hump is the norm rather than the exception.

At $Re = 9500$, Ta Phuoc Loc & Bouard's calculation does show the opening of the separating streamline and the subsequent interaction of the secondary eddy with the outer flow which was not the case at $Re = 3000$. It should be noted that they used a rather peculiar initial condition at both Reynolds numbers. At $T = 0$, their flow is initialized with a low Reynolds number solution[†] and the impulsive start is actually an abrupt change in the value of the viscosity. It is true that an impulsive start can be viewed as bringing the Reynolds number from infinity to a finite value. However, from a vorticity point of view, turning the viscosity off and turning the viscosity on are completely different situations. Immediately following the impulsive start, a singular vortex sheet diffuses into an otherwise irrotational fluid. In the Stokes solution, vorticity is already present in the fluid and the vorticity flux that immediately follows the abrupt rise in Reynolds number is the flux corresponding to steady solution, not the one associated with the irrotational flow.

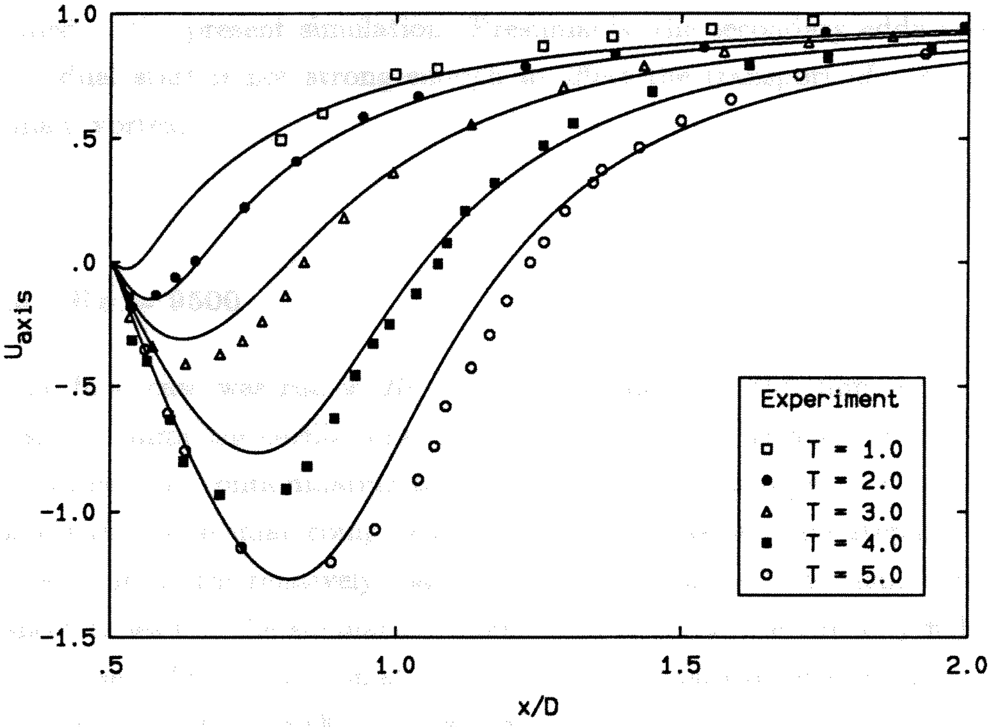


FIGURE 4.4.10 Comparison of the velocity on the symmetry axis for $Re = 3000$ obtained numerically (solid lines) with Bouard & Coutanceau experimental results.

[†] $Re = 5$.

By their own admission, Ta Phuoc Loc & Bouard chose this rather unphysical initial condition because it results in a better agreement with the experiments. Indeed, their recirculating velocities match the experimental values almost exactly while those of Fig. (4.4.10) show a significant discrepancy at $T = 3$ which corresponds to the formation of the new vortical structure.

It appears that the flow is very sensitive to its early history. The same conclusion was reached by Chamberlain (1988). He used a fourth order finite difference scheme to compare the wake generated by an impulsively started cylinder and the one left behind a cylinder given a finite and constant acceleration for 0.3 time units. The terminal velocity is the same in both case. Surprisingly, the recirculating velocities do not differ much at early times[‡] but exhibit different behaviors for $T > 3$. It appears that the flow never forgets its early history but instead, amplifies the slight discrepancies created by the different accelerations. The velocities computed by Chamberlain (op. cit.) with an impulsive start are actually very similar to those obtained in the present simulation. Presumably, the secondary eddy produced by the gradual start is not strong enough to affect the transport of vorticity toward the main vortex.

4.4.3 Re = 9500

The final case was run at $Re = 9500$. At this Reynolds number, there is less viscosity to damp the oscillations introduced by the numerical errors. To keep these instabilities from contaminating the solution, the overlap ratio, σ/h, is increased to 1.8 and a time step small compared to the optimum viscous time step is selected. As a consequence of the relatively coarse discretization, unrealistic results are expected for short times but the simulation should recover when the vorticity field becomes more diffused. The simulation at $Re = 9500$ was actually done with the same spatial resolution as for $Re = 3000$. This was also the case for Ta Phuoc Loc & Bouard's simulations.

It was first intended to carry the simulation up to $T = 4$, just like in the experiment. However, for the selected numerical parameters, the simulation of this

---

[‡] $T = 1$ and $T = 2$.

flow for times greater than $T \simeq 3.25$ involves more than 80,000 particles. The 32-node MarkIII simply cannot handle that many particles and the simulation is stopped at that point.

$N_{\max}$	σ	σ/h	No. of panels	Δt	$\Delta t/\Delta t_{\text{opt}}$
79672	0.007	1.8	2216	0.018	0.155

TABLE 4.4.3 Numerical parameters for the simulation at $Re = 9500$.

Again, the reader is referred to App. **B** for a complete description of the wake development. The events that characterize this development are very similar to those observed at $Re = 3000$. The major difference is that the separating vorticity layer rolls up sooner and closer to the cylinder resulting in a compact structure that really dominates the recirculating flow. The emergence of this structure also causes the secondary separation at $T \simeq 1.83$ and eventually produces the characteristic β-phenomenon at $T \simeq 2.25$. One feature of that phenomenon is not reproduced numerically: namely the relative isolation of the fluid located between the main vortical structure and the symmetry axis. In the simulation, this region appears to be strongly coupled with the primary eddy.

The secondary eddy splits the main recirculation region at $T \simeq 2.34$ and stays in direct communication with the outer flow until $T \simeq 2.68$. As in the previous simulation, the bridge connecting the primary separation point and the main eddy is ruptured during that period and the vorticity flux is redirected toward the cylinder. It is the accumulation of vorticity in this new structure that causes the weakening of the secondary eddy and eventually re-opens the path toward the main vortex.

Figures (4.4.11) and (4.4.12) provide a comparison of the numerical simulation with a flow visualization obtained at the same dimensionless time. The flow features are qualitatively similar but the agreement with Bouard & Coutanceau's pathlines is not as good as for the lower Reynolds number.

The agreement is actually better when the experimental pathlines are compared

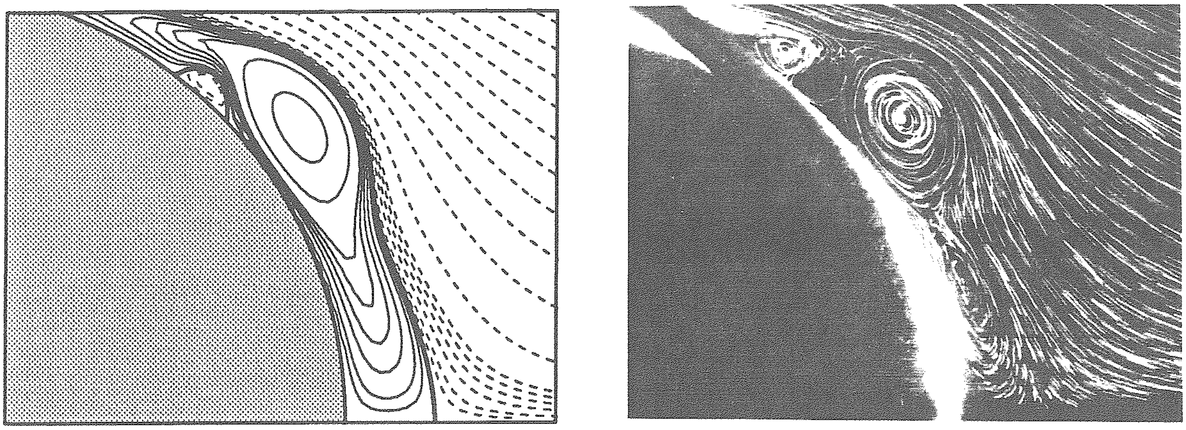


FIGURE 4.4.11 Comparison of computed streamlines with Bouard & Coutanceau experimental flow visualization at $Re = 9500$ and $T = 2.0$.

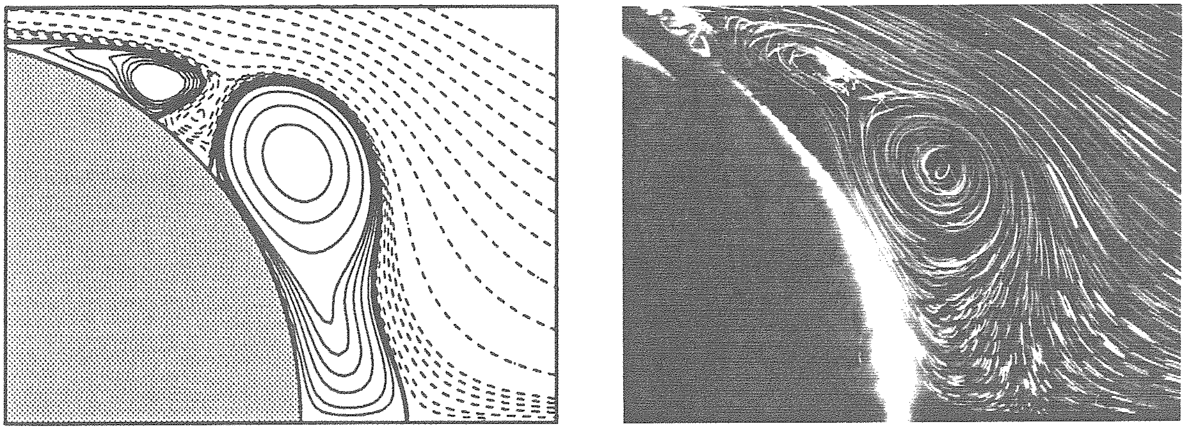


FIGURE 4.4.12 Comparison of computed streamlines with Bouard & Coutanceau experimental flow visualization at $Re = 9500$ and $T = 2.5$.

with the streamlines computed at a slightly later time*. There is also a striking similarity between the streamlines computed for $T = 3.0$ and Ta Phuoc Loc & Bouard's simulation at $T = 2.8$. It seems that by dominating the problem at the

---

* Approximately .25 time units.

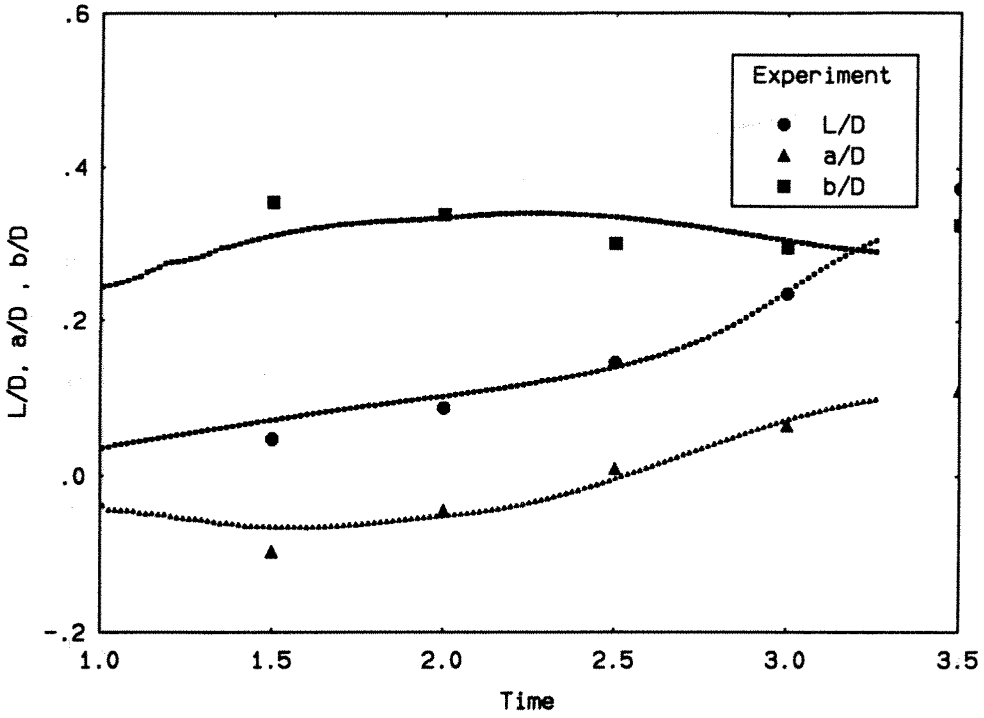


FIGURE 4.4.13 Comparison of the time history of the computed geometrical parameters of the closed wake for $Re = 9500$ with Bouard & Coutanceau experimental measurements

early times, the numerical viscosity has delayed the wake development.

The time history of the wake geometrical parameters is shown on Fig. (4.4.13). Overall, the computation tracks the development of the wake fairly accurately, especially at the later times. Considering the difficulties encountered at the early stages of the simulation, this agreement says more about the robustness of the diagnostics based on the streamfunction than about the quality of the simulation.

The progression of the separation angle along the rear portion of the cylinder is shown on Fig. (4.4.14). The first signs of separation are observed at roughly the same time as for $Re = 3000$, namely $T \simeq 0.52$. This was not totally unexpected since Bar-Lev & Yang matched asymptotics show very little variation of the separation time in that range of Reynolds numbers. However, their analysis predicts the appearance of a recirculating flow as soon as $T \simeq 0.38$. A simpler boundary layer analysis applied to the $2\sin\theta$ irrotational velocity distribution gives $T = .35$ as the onset of separation. This analysis does not involve the Reynolds number and

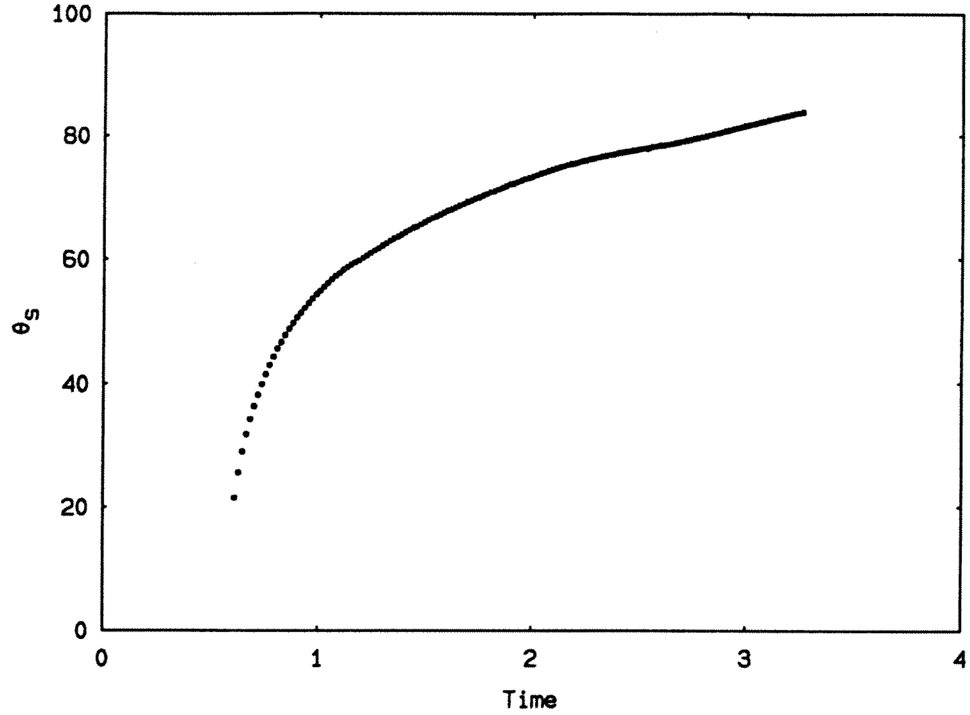


FIGURE 4.4.14 Time history of separation angle for $Re = 9500$.

the predicted separation time can be viewed as a lower bound, valid at very high Reynolds numbers.

This delay is not simply the result of a coarse discretization but mostly shows the inadequacy of the strategy used to identify the numerical separation time. Separation always takes place at the rear stagnation point and is imminent when

$$\left.\frac{\partial\omega}{\partial\theta}\right|_{\substack{r=R\\ \theta=0}} = 0 \ . \tag{4.4.1}$$

That definition was used by Bar-Lev & Yang and by Blasius (1908) in his boundary layer analysis. In the proposed vortex method, all particles lie within the fluid and the vorticity field is poorly represented at the wall itself. Eq. (4.4.1) cannot be used and instead, the streamlines are closely examined in order to detect the emergence of a recirculation zone. At these early times, the flow does not evolve rapidly and it seems that Eq. (4.4.1) is satisfied long before separation manifests itself in the streamfunction. As a result, the use of the proposed scheme, and of vortex methods in general, is not recommended when an accurate prediction of the separation time is sought.

Finally, Fig. (4.4.15) shows that the bursting of the secondary eddy into the outer flow ($T \simeq 2.5$) slows down the progression of the separation angle just as it did at $Re = 3000$.

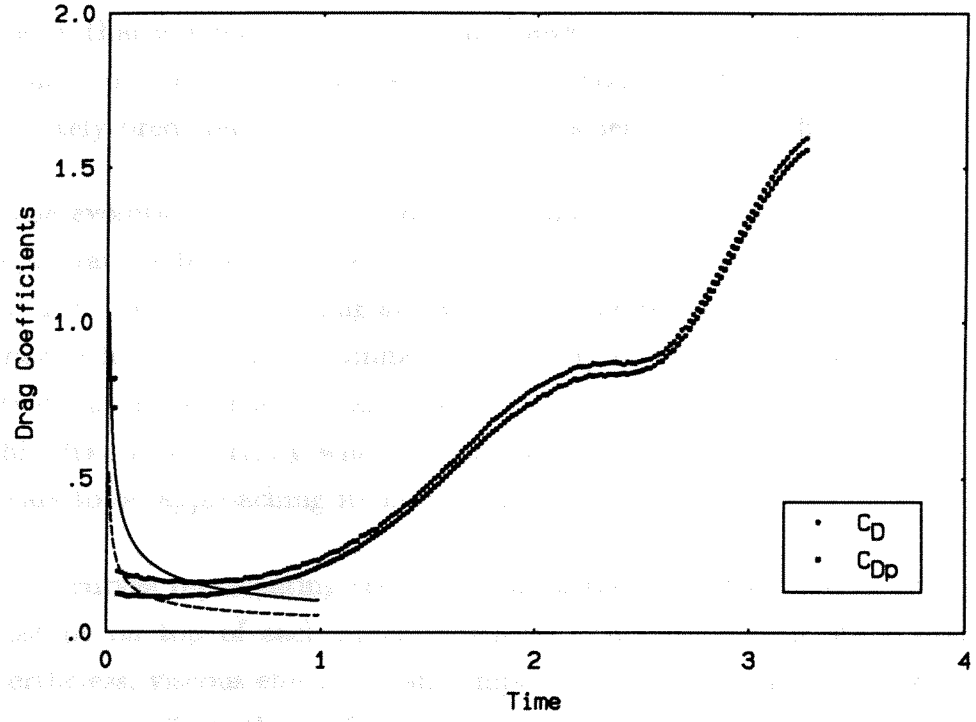


FIGURE 4.4.15 Time history of the computed drag coefficients for $Re = 9500$ compared with Bar-Lev & Yang asymptotic results (solid and dash lines).

The drag curves shown on Fig. (4.4.15) are typical of an under-resolved simulation. In that regard, they are even worse than the ones associated with the coarsest case presented in the convergence study. Not only is the spatial resolution too coarse for the early vorticity gradient but it is also too large with respect to the time step. As it leaves the walls, the vorticity flux is distributed over the regularization length σ, which also defines the spatial resolution. In the simulation presented in this subsection, σ is larger than the diffusive length over which the newly created vorticity is expected to travel during a time step. In other words, the optimum viscous time step is large compared with the selected time step. As a result, vorticity is artificially transported away from the wall and the drag coefficients are unrealistically predicted for short times where the flow is dominated by the near-wall activity.

This is a situation where a larger time step would have produced a better asymptotic behavior near $T = 0$. Unfortunately, a small time step is needed to keep the simulation stable. For $T > 1$, while some of the flux is still deposited too far from the wall, the evolution of the drag is mostly determined by the behavior of the vorticity that is already in the fluid and away from the cylinder. The coarsest case presented in the convergence study has demonstrated that this behavior can be adequately predicted despite the numerical smearing of the flux.

The evolution of the profile drag is similar to what was observed at $Re = 3000$ except that the feeding of the primary vortex is interrupted sooner. The drag levels off at $T \simeq 2$ when the feeding and growth of the primary eddy is interrupted. This primary eddy is still in the immediate vicinity of the cylinder when the separating vorticity layer can reach it again and a dramatic drag increase is observed. The profile drag is still rising when the simulation is stopped at $T \simeq 3.27$ although it appears to be approaching its maximum value.

The curves representing the evolution of the total drag and of the profile drag almost lie on top of each other indicating that the viscous shear drag is small. Nevertheless, viscous effects are still important in determining the separation point which in turn affects the profile drag. Actually, by rolling up more tightly and closer to the cylinder, the separating vortex sheet produces a larger total drag coefficient than at the previous Reynolds numbers. It does so despite the lesser shear drag.

It should also be noted that the effects of remeshing are more evident on Fig. (4.4.15) than at any other Reynolds number. The violent flow resulting from the more intense vorticity distributed over smaller length scales severely deforms the Lagrangian mesh. Not only is it more difficult to conserve the fluid linear impulse while remeshing, but when the impulse is differentiated with respect to time, the small time step used in this simulation amplifies the small variations due to remeshing.

The component of the velocity along the symmetry axis is plotted on Fig. (4.4.16) in the region immediately behind the cylinder. For short times, the fatter wake produced by the numerical smearing results in large discrepancies in the velocity field. However, it appears that the under-resolved simulation eventually recovers

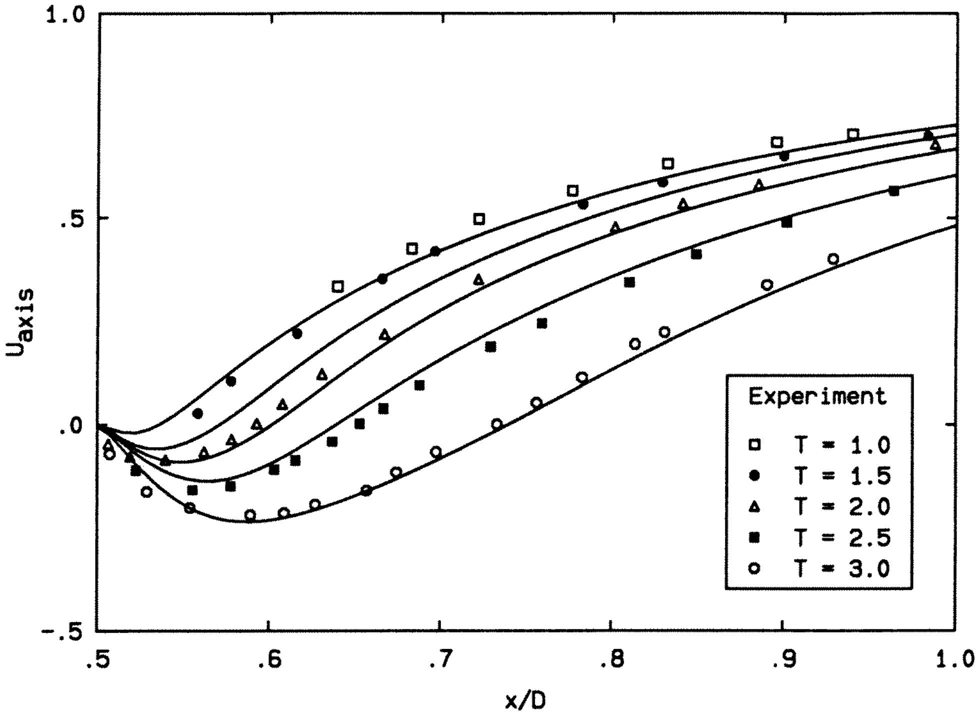


FIGURE 4.4.16 Comparison of the velocity on the symmetry axis for $Re = 9500$ obtained numerically (solid lines) and experimentally.

from its shortcomings and the numerical results for $T \geq 2$ compare surprisingly well with the experimental values. Unfortunately, the simulation had to be stopped at $T = 3.27$ and it was not possible to determine if that trend would have persisted.

CHAPTER 5

Conclusions

It was shown in Chap. 2 that the diffusion of vorticity can be modeled by an exchange of circulation between discrete Lagrangian particles. The integral representation of the Laplacian, first proposed by Mas-Gallic (1985) and Raviart (1985, 1987) for unbounded flows, was extended to situations where a flux of vorticity is used to enforce the no-slip condition on solid surfaces. Used in tandem with a traditional inviscid vortex method, this numerical scheme was applied to the simulation of the symmetric flow past an impulsively started cylinder.

The results shown in Chap. 4 confirm the soundness of this approach. For a range of Reynolds numbers that extends at least from 550 to 3000, the proposed scheme was capable of resolving all the significant length and time scales present in the problem. The evidence is twofold. First, the simulations were able to reproduce the interaction of the separated flow with the rear portion of the cylinder. Under some circumstances, the recirculating flow can itself separates from the cylinder's surface. The numerical scheme captured the dependence of the secondary phenomena on the Reynolds number as observed in Bouard & Coutanceau's experiments (1980). A barely noticeable tertiary separation was even correctly predicted. The second piece of evidence is related to the drag experienced by the cylinder immediately after the impulsive motion. For the first few time steps, the computed drag coefficients* hug the singular behavior predicted analytically by Bar-Lev & Yang (1975) using matched asymptotic expansions. The steepest vorticity gradients occur at the very early stages of the wake development. The confirmation that the numerical scheme can accurately handle these gradients greatly enhances our confidence that it can deal with the smoother vorticity field present at later times.

---

* Total and form drag.

The presence and behavior of a secondary phenomenon were mainly used to validate the numerical scheme. The simulations have also shown that the secondary eddy is more than a cosmetic rearrangement of the flow field. When it has grown to the point where it bursts into the outer flow, the secondary eddy significantly affects the convective transport of vorticity. Instead of reaching the main vortical structure, the vorticity released at the primary separation point is redirected toward the cylinder and the separating vortex layer rolls up into a new vortical structure. Because of its location, the new structure is not as effective a drag producer as the primary roller. Consequently, the reorganization of the vorticity field leaves a characteristic signature on the time history of the drag coefficients: a momentary reduction in the drag experienced by the cylinder is always associated with the bursting of the secondary eddy.

The interaction of the secondary eddy with the outer flow has been observed at $Re = 3000$ as well as for $Re = 9500$. However, at Chap. 4's highest Reynolds number, $Re = 9500$, the simulation is clearly under-resolved and the behavior of the drag at short times does not match the analytical result. The comparison of the computed streamlines with Bouard & Coutanceau's flow visualizations is also less than convincing. This is only a matter of resolution and, presumably, faster and larger computers could produce a simulation that is as successful as the ones obtained within the window: $550 \leq Re \leq 3000$. At this point, it should be noted that Chap. **3**'s combination of fast algorithm and concurrent processing was instrumental in creating that window. Simulations made with a number of vortices reduced by an order of magnitude** would have been suspect, even at $Re = 550$.

Besides yielding results that compare well with experiments, the proposed scheme is appealing because it removes most of the hand waving that has been associated with the vortex simulation of bluff body flows. It does not involve boundary layer approximations, arbitrary separation points and the like. Unfortunately, the proposed scheme did not completely succeed in taking the ad hockery out of viscous vortex methods. An accurate description of the vorticity field is required to properly model the viscous transport of vorticity by an exchange of circulations. That accuracy cannot be maintained when the Lagrangian mesh is distorted to the point where the particle cores no longer overlap. The simulation has to be periodically rescued by projecting the vorticity field known from a distorted set of

---

** A typical number for an $\mathcal{O}(N^2)$ simulation on a CRAY.

particles onto a new set that is uniformly distributed. Remeshing is an empirical, but necessary, ingredient of the simulation of bluff body flows using the proposed numerical scheme.

Since the viscous operator is written for a constant regularization length, σ, the overlap requirement also limits the type of problem that can be successfully simulated. In practice, the spatial resolution is the same throughout the discretized domain and the computational effort spent on a blob of fluid distant from the cylinder is the same as for one in immediate contact with the wall. As more fluid becomes rotational, this lack of focus rapidly makes the method prohibitively expensive, even within a fast algorithm framework. The uniform spatial resolution is tolerable only when the vorticity field is confined to the immediate vicinity of the solid body. Such flows can be created by a start-up process, like the ones computed in Chap. 4, or by an oscillatory motion of the solid body.

Because of this restriction, an advantage of vortex methods was left untapped in this study, namely, the treatment of the boundary condition at infinity. Unlike Eulerian schemes, vortex methods do not require an outer boundary condition. However, in the finite differences computation of the impulsively started cylinder, the outer boundary is far enough from the body to avoid any interaction of the wake with the condition at "infinity," and a truly infinite domain is not really an advantage.

Nevertheless, vortex methods coupled with a "deterministic" treatment of the diffusive term now offer a potent alternative to more conventional approaches like finite differences for the simulation of the early stages of the wake development behind a bluff body.

Recommendations for further investigations

Without significantly modifying the method, other interesting problems could be considered within the symmetric compact wake constraint. For example, the shape of cylinder generator could be modified in an attempt to manipulate the behavior of the primary vortices. The drag could be reduced by facilitating the drifting of the primary eddies. A fin, acting as an extra source of counter-rotating vorticity, could even be added to the cylinder. In that case, and in any problem involving a sharp edge, the weight redistribution method would have to be modified to prevent the diffusion of vorticity across a solid surface. This is not necessary for the circular cylinder as the smoothing length, σ, is always much smaller than the cylinder's radius of curvature.

Relaxing the symmetry requirement would increase the cost of a given simulation by a factor of two and further reduce the range of Reynolds number where the method can be applied with confidence. However, such a scheme could compute the early development of a wake left behind a lifting body such as an airfoil or even a spinning cylinder. For both problems, experimental observations (see Coutanceau & Ménard (1985) and Ohmi, Coutanceau, Phuoc Loc & Dulieu (1990)) could be used to build confidence in the numerical method.

References

Abramowitz, M. & Stegun, I.E. 1972. *Handbook of Mathematical Functions, (Tenth Printing)*. National Bureau of Standards.

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.

Appel, A. 1985. An efficient program for many-body simulation. *SIAM J. Sci. Stat. Comput.* **6**, 85.

Baden, S.B. 1987. *Run-time partitioning of scientific continuum calculations running on multiprocessors*. Ph. D. thesis, University of California, Berkeley.

Bar-Lev, M. & Yang, H.T. 1975. Initial flow field over an impulsively started circular cylinder. *J. Fluid Mech.* **72**, 625.

Barnes, J. & Hut, P. 1986. A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm. *Nature* **324**, 446.

Barnes, J. 1990. A modified tree code: Don't laugh; it runs. *J. Comp. Phys.* **87**, 161.

Bateman, H. 1954. *Tables of Integral Transforms*. Vol. 2, McGraw-Hill.

Beale, T. & Majda, A. 1981. Rates of convergence for viscous splitting of the Navier-Stokes equations. *Math. Comp.* **156**, 243.

Bingham, H.H., Weimer, D.K. & Griffith, W. 1952. The cylinder and semicylinder in subsonic flow. *Princeton University, Department of Physics, Technical Report 11-13*.

Birdsall, C.K. & Fuss, D. 1969. Clouds-in-cloud, clods-in-cell physics for many-body plasma simulation. *J. Comp. Phys.* **3**, 494.

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

Bouard, R. & Coutanceau, M. 1980. The early stage of development of the wake behind an impulsively started cylinder for $40 \leq \mathrm{Re} \leq 10^4$. *J. Fluid Mech.* **101**, 583.

Bracewell, R.M. 1965. *The Fourier transform and its applications.* McGraw-Hill.

Bryson, A.E. 1959. Symmetric vortex separation on circular cylinder and cones. *J. Appl. Mech. Trans. ASME, Ser. E* **11**, 643.

Chamberlain, R.R. 1988. Unsteady flow phenomena in the near wake of a circular cylinder. *AIAA J.* **26**, 8.

Cheer, A.Y. 1983. Numerical study of incompressible slightly viscous flow past blunt bodies and airfoils. *SIAM J. Sci. Comput.* **4**, 685.

Cheer, A.Y. 1989. Unsteady separated wake behind an impulsively started cylinder in slightly viscous flow. *J. Fluid Mech.* **201**, 485.

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

Chorin, A.J. 1978a. A comment on the paper "The calculation of large Reynolds number two-dimensional flow using discrete vortices with random walk" by F. Milinazzo & P.G. Saffman. *J. Comput. Phys.* **26**, 453.

Chorin, A.J. 1978b. Vortex sheet approximation of boundary layers. *J. Comput. Phys.* **27**, 428.

Christiansen, J.P. 1973a. *The non-linear dynamics of vortex flows by numerical methods.* Ph. D. thesis, Warwick University.

Christiansen, J.P. 1973b. Numerical simulation of hydromechanics by the method of point vortices. *J. Comput. Phys.* **13**, 363.

Cottet, G.H. 1989. A particle-grid superposition method for the Navier-Stokes equations. To be published.

Coutanceau, M. & Bouard, R. 1977. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 2. Unsteady flow. *J. Fluid Mech.* **79**, 257.

Coutanceau, M. & Ménard, C. 1985. Influence of rotation on the near wake development behind an impulsively started circular cylinder. *J. Fluid Mech.* **158**, 399.

Daube, O. & Ta Phuoc Loc 1978. Etude numérique d'écoulement instationnaires de fluide visqueux incompressible autour de corps profilé par une méthode combinée d'ordre $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$. *J. Méc.* **17**, 651.

Degond, P. & Mas-Gallic, S.. The weighted particle method for convection-diffusion equations, part I: the case of isotropic viscosity. To appear in *Math. Comput.*.

Degond, P. & Mustieles, F.-J. 1990. A deterministic approximation of diffusion using particles. *SIAM J. Sci. Stat. Comput.* **11**, 293.

Fishelov, D. 1990. A new vortex scheme for viscous flows. *J. Comput. Phys.* **86**, 211.

Fox, G., Johnson, M., Lyzenga, G., Otto, S., Salmon, J. & Walker, D. 1988. *Solving problems on concurrent processors*. Prentice-Hall.

Gerrard, J.H. 1967. Numerical computation of the magnitude and frequency of the lift on a circular cylinder. *Phil. Trans. Roy. Soc.* **261**, 137.

Giesing, J.P. 1965. Potential flow about two-dimensional airfoils. *Douglas report LB31946*.

Goldstein, S. & Rosenhead, L. 1936. Boundary layer growth. *Proc. Camb. Phil. Soc.* **32**, 392.

Gradshteyn, I.S. & Ryzhik, I.M. 1980. *Table of Integrals,Series and Products*. Academic Press.

Gray, A. & MacRobert, T.M. 1931. *A treatise on Bessel functions and their applications to physics*. MacMillan.

Greengard, C. 1985. The core spreading vortex method approximates the wrong equation. *J. Comput. Phys.* **61**, 345.

Greengard, L. & Rokhlin, V. 1987. A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325.

Hald, O.H. 1979. Convergence of vortex methods for Euler's equations. *SIAM J. Num. Analysis* **16**, 726.

Hockney, R., Goel, S.P. & Eastwood, J. 1974. Quiet high resolution computer model of a plasma. *J. Comput. Phys.* **14**, 148.

Hockney, R. & Eastwood, J. 1981. *Computer simulation using particles.* McGraw-Hill.

Ingham, D.B. 1968. Note on the numerical solution for unsteady viscous flow past a circular cylinder. *J. Fluid Mech.* **31**, 815.

Inoue, O. & Leonard, A. 1987. Vortex simulation of forced/unforced mixing layers. *AIAA J.* **25**, 1417.

Katzenelson, J. 1989. Computational structure of the N-body problem. *SIAM J. Sci. Stat. Comput.* **11**, 787.

Knuth, D. E. 1986. *The TEXbook : Computers & typesetting.* Addison Wesley.

Lebedev, N.N. 1963. *Special Functions and their Applications.* Dover Publications.

Lecointe, Y. & Piquet, J. 1984. On the use of several compact methods for the study of the incompressible viscous flow around a circular cylinder. *Comp. & Fluids* **12**, 225.

Lecointe, Y. & Piquet, J. 1985. Unsteady viscous flow round moving circular cylinder and airfoils. *AIAA-85-1490.*

Leonard, A. 1980. Vortex methods for flow simulation. *J. Comput. Phys.* **37**, 289.

Leonard, A. 1985. Computing three-dimensional incompressible flows with vortex elements. *Ann. Rev. Fluid Mech.* **17**, 523.

Mas-Gallic, S. & Raviart, P.A. 1985. Particle approximation of convection-diffusion problems. *Report, Université de Paris 6, Paris, France.*

McCroskey, W.J. 1977. *Some current research in unsteady fluid dynamics - The 1976 Freeman Scholar lecture.* J. Fluid Eng. **99**, I, 8.

Milinazzo, F. & Saffman, P.G. 1977. The calculation of large Reynolds number flow using discrete vortices with random walk. *J. Comp. Phys.* **23**, 380.

Monaghan, J.J. 1982. Why particle methods work. *SIAM J. Sci. Stat. Comput.* **3**, 422.

Monaghan, J.J. & Gingold, R.A. 1983. Shock simulation by a particle method SPH. *J. Comput. Phys.* **52**, 374.

Nagata, H., Funada, H., Kawai, K. & Matsui, T. 1985. Unsteady flows in the vortex region behind a circular cylinder started impulsively. *Bulletin of JSME* **28**, 2599.

Nagata, H., Nagase, I. & Ito, K. 1989. Unsteady flows past a circular cylinder started impulsively in the Reynolds number range $500 < Re < 10^4$. *JSME Int. Journal* **32**, 540.

Nakayama, Y. 1988. *Visualized flow; fluid motion in basic and engineering situations revealed by flow visualization.* Pergamon Press.

Ohmi, K., Coutanceau, M., Ta Phuoc Loc & Dulieu, A. 1990. Vortex formation around an oscillating and translating airfoil at large incidence. *J. Fluid Mech.* **211**, 37.

Patel, V.A. 1976. Time dependant solutions of the viscous incompressible flow past a cylinder. *Comp. Fluids* **4**, 13.

Prandtl, L. 1904. Über Flussigkeitsbewegung mit kleiner Reiburg. Engl. trans., *NACA TM-452*.

Prandtl, L. & Tietjens, O.G. 1934. *Applied hydro and aeromechanics*. McGraw-Hill.

Raviart, P.A. 1985. *An analysis of particle methods.* In Numerical methods in fluid dynamics, F. Brezzi, ed., Lecture notes in Mathematics 1127, Springler-Verlag.

Raviart, P.A. 1987. *Méthodes particulaires.* Lecture notes, Ecole d'été d'analyse numérique, Centre d'étude du Bréau-sans-nappe, France.

Rosenhead, L. 1931. The formation of vortices from a surface of discontinuity. *Proc. Roy. Soc., Ser. A* **134**, 170.

Saffman, P.G. 1981. Dynamics of vorticity. *J. Fluid Mech.* **106**, 49.

Salmon, J. 1984. An astrophysical N-body simulation for hypercubes. *Caltech Concurrent Computation report no 78.*

Sarpkaya, T. 1968. An analytical study of separated flow about circular cylinders. *J. Basic Eng.* **90**, 511.

Sarpkaya, T. 1989. *Computational methods with vortices - The 1988 Freeman Scholar lecture.* J. Fluid Eng. **111**, I, 5.

Schwabe, M. 1943. Pressure distribution in non uniform two-dimensional flow. *NACA TM-1043.*

Schuh, H. 1953. Calculation of unsteady boundary layers in two dimensional laminar flow. *Z. Flugwiss.* **1**, 122.

Smith, P.A. & Stansby, P.K. 1988. Impulsively started flow around a circular cylinder by the vortex method. *J. Fluid Mech.* **194**, 45.

Smith, P.A. & Stansby, P.K. 1989. Postcritical flow around a circular cylinder by the vortex method. *J. of Fluids & Structures* **3**, 275.

Son, J.S. & Hanratty, T.J. 1969. Numerical solution of the flow around a cylinder at Reynolds number of 40, 200, 500. *J. Fluid Mech.* **35**, 369.

Spalart, P.R. & Leonard, A. 1981. Computation of separated flows by a vortex-tracing algorithm. *AIAA-81-1246.*

Spalart, P.R. 1982. *Numerical simulation of separated flows.* Ph. D. thesis, Stanford University.

Spalart, P., Leonard, A. & Baganoff, D. 1983. Numerical simulation of separated flows. *NASA-TM-84328.*

Stewartson, K., Belcher, B.J., Burggraf, O.R., Cooke, J.C. & Robins, A.J. 1971. Limit-less boundary layers. *In Recent research on unsteady boundary layer, ed. Eichelbrenner, E.A.*, 1444.

Ta Phuoc Loc 1980. Numerical analysis of unsteady secondary vortices generated by an impulsively started circular cylinder. *J. Fluid Mech.* **100**, 111.

Ta Phuoc Loc & 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.

Thom, A. 1933. The flow past circular cylinders at low speeds. *Proc. Roy. Soc., A* **141**, 651.

Tiemroth, E.C. 1986. *The simulation of the viscous flow around a cylinder by the random vortex method.* Ph. D. thesis, University of California, Berkeley.

Wang, C.Y. 1967. The flow past a circular cylinder which is started impulsively from rest. *J. Maths & Phys.* **46**, 195.

Winckelmans, G. & Leonard, A. 1988. Improved vortex methods for three dimensional flows. *Proceedings of the SIAM Workshop on Vortex Dynamics*, Leesburg,Virginia.

Winckelmans, G. 1989. *Topics in vortex methods for the computation of three and two dimensional incompressible unsteady flows*. Ph. D. thesis, Caltech.

APPENDIX A

Numerical integration of the flux redistribution term

This appendix is concerned with the evaluation of the boundary term appearing in Eq. (2.1.18), which is rewritten here as

$$I(x_i, y_i) = h^2 \int_\Gamma R_\sigma \left(|\mathbf{x}_i - \mathbf{x}_\Gamma| \right) \nu \frac{\partial \omega(\mathbf{x}_\Gamma)}{\partial n} \, d\mathbf{x}_\Gamma \tag{A.1}$$

This integral has to be evaluated at every particle location, (x_i, y_i), in order to determine what fraction of the flux will be allocated to this specific particle. The flux itself is not known analytically; it is found by cancelling the observed slip velocity at a limited number of control points. It will be assumed that the flux, F_ω, is constant within each panel. To further simplify the integral, the boundary itself is represented by either straight panels or circular arcs. Analytical solutions were sought for both regularization kernels with increasing levels of sophistication in the approximation of the boundary.

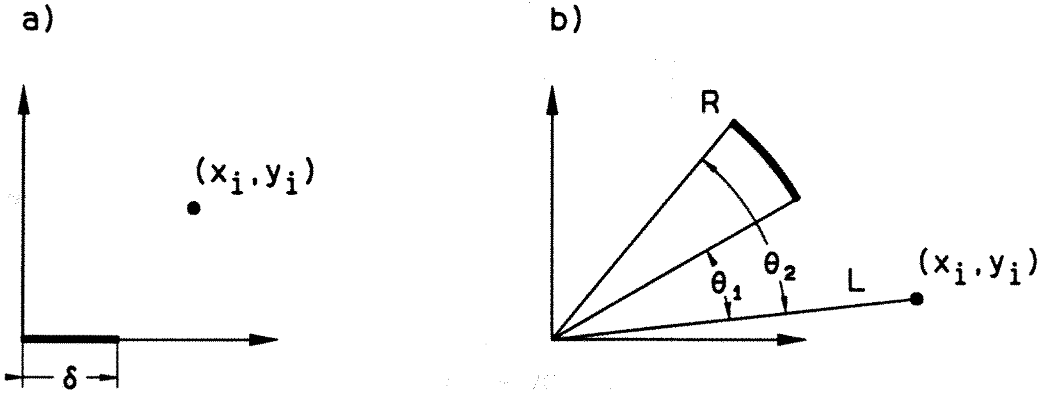


FIGURE A.1 Panel geometry for a), a straight panel and for b), a circular panel.

For a Gaussian regularization, Eq. (A.1) has a closed form solution for the straight panel only, which can be expressed as

$$I(x_i, y_i) = \frac{F_\omega}{2} \frac{h^2}{\sqrt{2\pi\sigma^2}} \exp(-\frac{y_i^2}{2\sigma^2}) \operatorname{erf}\left(\frac{x - x_i}{\sqrt{2\sigma^2}}\right)\Bigg|_{x=0}^{x=\delta} \quad . \tag{A.2}$$

A solution for the algebraic regularization defined by Eq. (2.2.2) can be found for both type of geometries. If the panel is straight, then

$$\begin{aligned} I(x_i, y_i) = {} & F_\omega \frac{h^2\sigma^6}{\pi} \left(\frac{x - x_i}{1 + y_i^2}\right) \left[\frac{1}{2P(x)^2} + \frac{3}{4(1 + y_i^2)P(x)}\right]_{x=0}^{x=\delta} \\ & + \frac{3F_\omega}{4\pi} \frac{1}{(1 + y_i^2)^{\frac{5}{2}}} \arctan\left(\frac{x - x_i}{\sqrt{1 + y_i^2}}\right)\Bigg|_{x=0}^{x=\delta} \end{aligned} \tag{A.3}$$

where

$$P(x) = x^2 - 2x_i x + x_i^2 + y_i^2 + 1 \quad . \tag{A.4}$$

If the panel is a circular arc of radius R, the algebraic regularization produces

$$\begin{aligned} I(x_i, y_i) = {} & F_\omega \frac{h^2\sigma^6}{2\pi} \left[\frac{\beta \sin\theta}{2(\alpha^2 - \beta^2)(\alpha + \beta\cos\theta)^2}\right]_{\theta=\theta_1}^{\theta=\theta_2} \\ & + \frac{1}{2(\alpha^2 - \beta^2)^2} \left[\frac{3\alpha\beta\sin\theta}{(\alpha + \beta\cos\theta)} \right. \\ & \left. - \frac{2(2\alpha^2 + \beta^2)}{\sqrt{\alpha^2 - \beta^2}} \arctan\left(\frac{\sqrt{\alpha^2 - \beta^2}\tan(\theta/2)}{\alpha + \beta}\right)\right]_{\theta=\theta_1}^{\theta=\theta_2} \quad , \end{aligned} \tag{A.5}$$

where

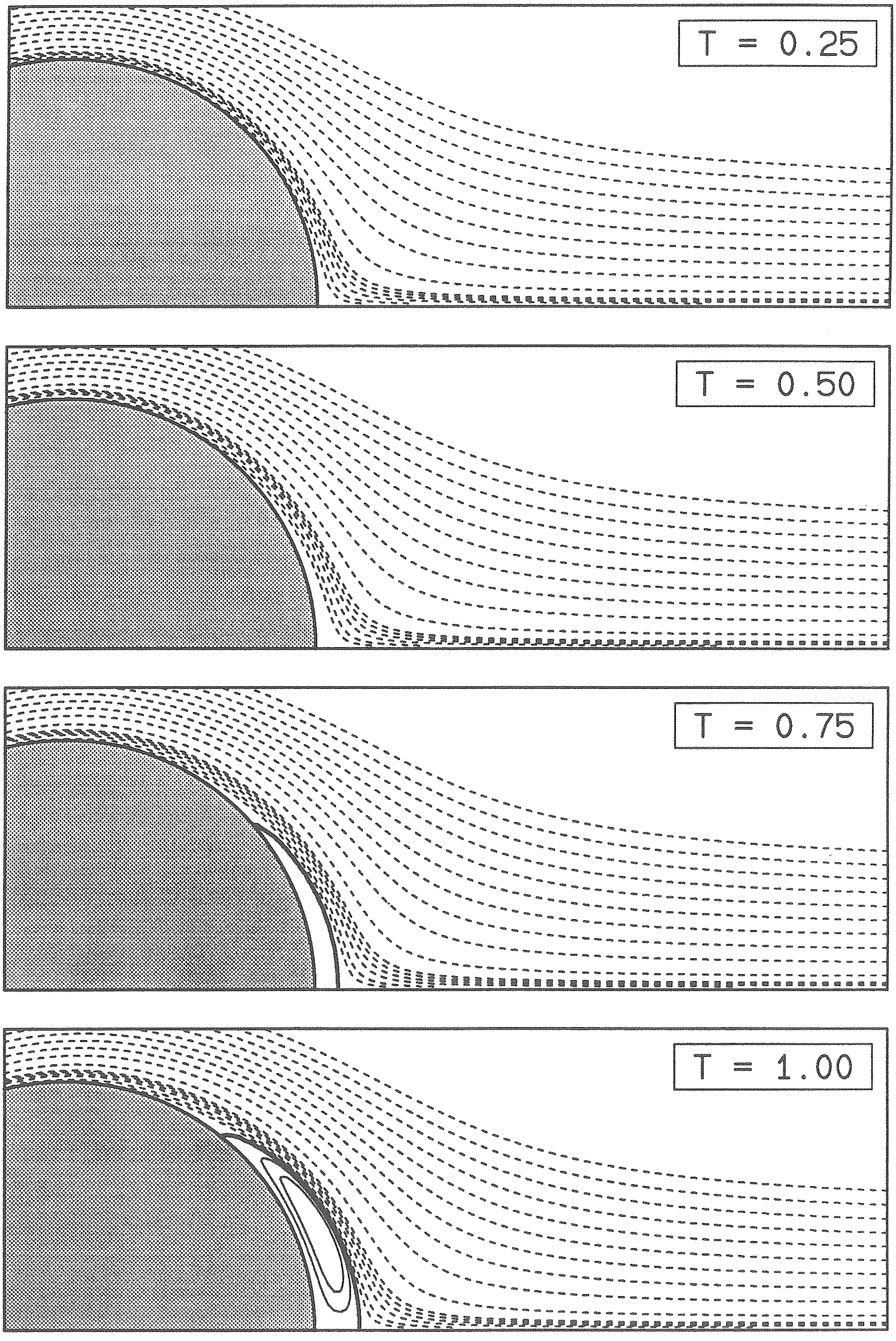
$$\begin{aligned} L &= \sqrt{x_i^2 + y_i^2} \\ \alpha &= 1 + R^2 + L^2 \\ \beta &= -2RL \quad . \end{aligned} \tag{A.6}$$

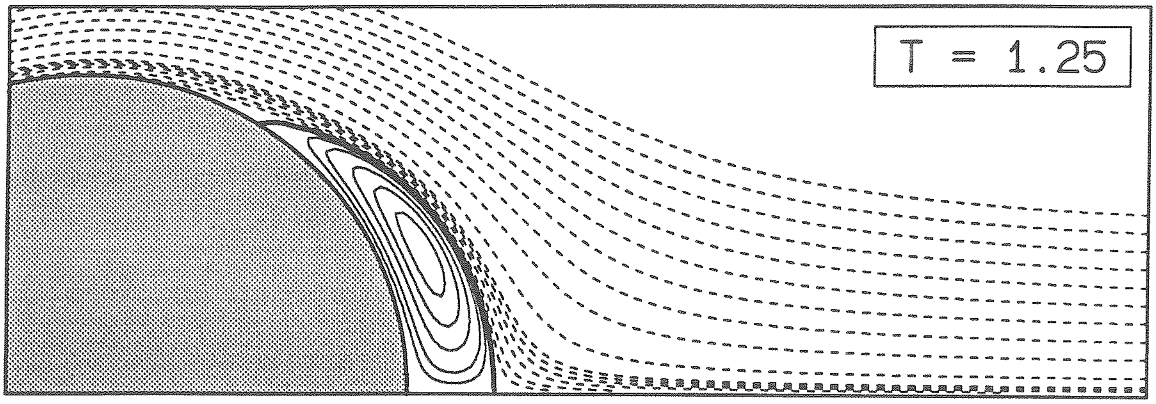
APPENDIX B

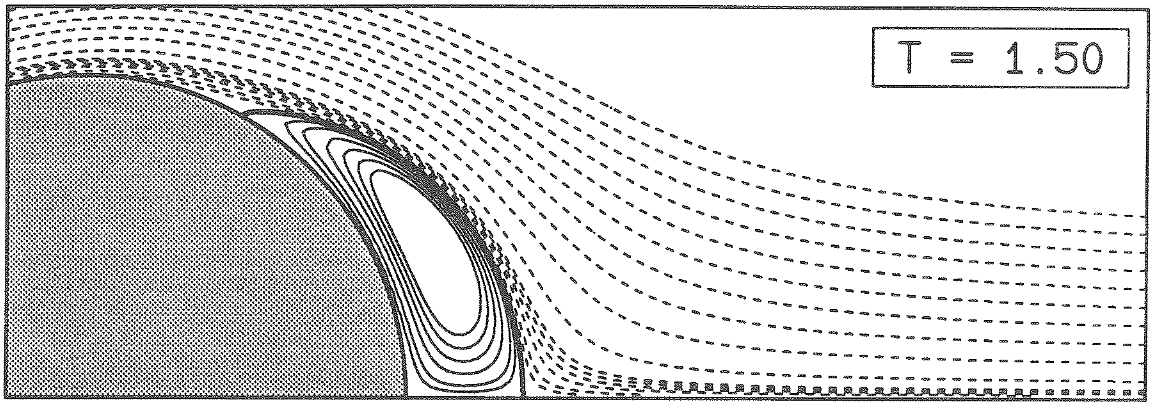
Flow patterns around an impulsively started cylinder

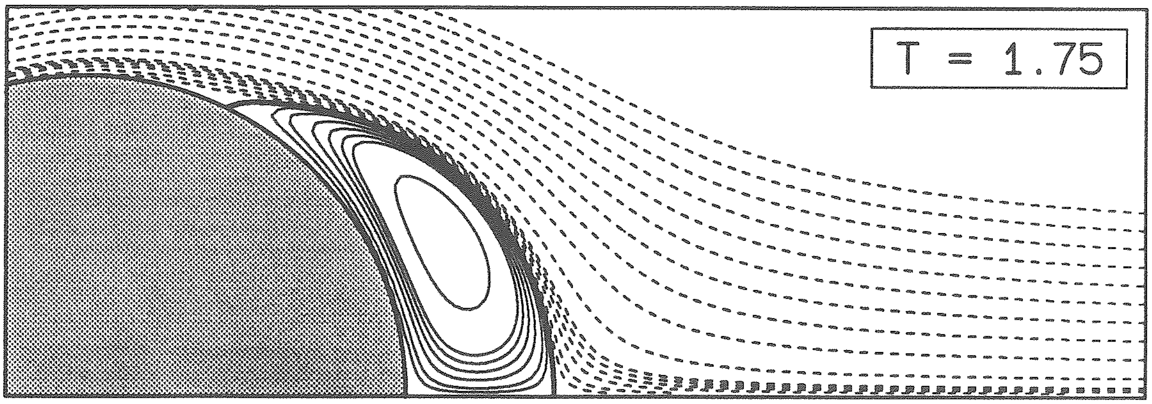
This appendix is concerned by the flow patterns generated by the impulsive start of a circular cylinder at $Re = 550$, 3000 and 9500. Snapshots of the streamfunction are shown at regular intervals ($\Delta T = 0.25$) using the following convention. Contours are plotted at regular increment of 0.05 between $\psi = -0.5$ and $\psi = 0.5$. The streamline $\psi = 0$ divides the closed wake from the outer flow and is bolder than the other streamlines to draw attention to its special nature. In order to discern more details near the dividing streamline, twelve additional contours are shown at regular interval from $\psi = -0.03$ to $\psi = 0.03$. Negative contours are drawn with solid lines while dashes are used for positive values.

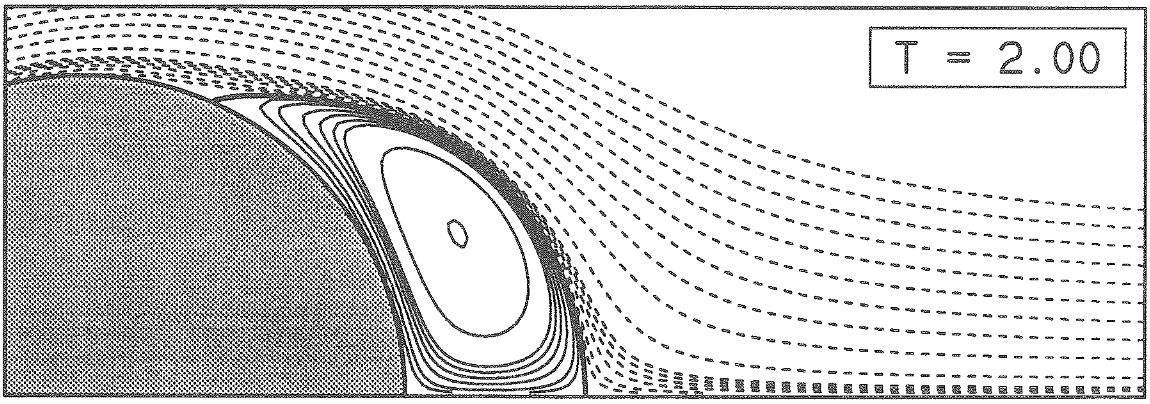
$\mathbf{Re = 550}$

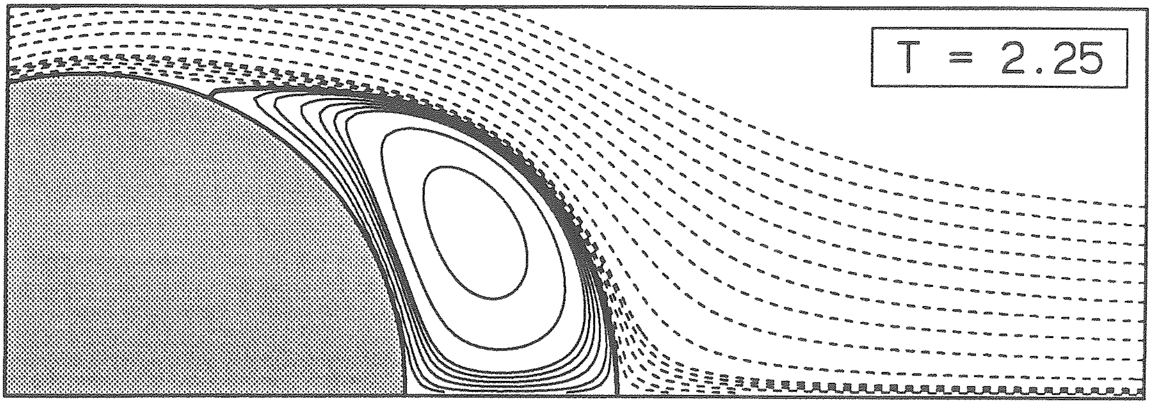

T = 0.25
T = 0.50
T = 0.75
T = 1.00

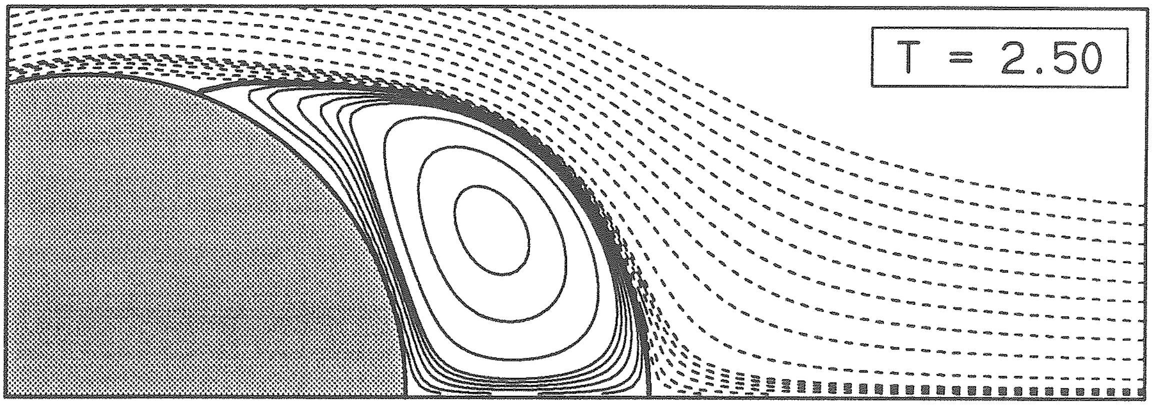

T = 1.25

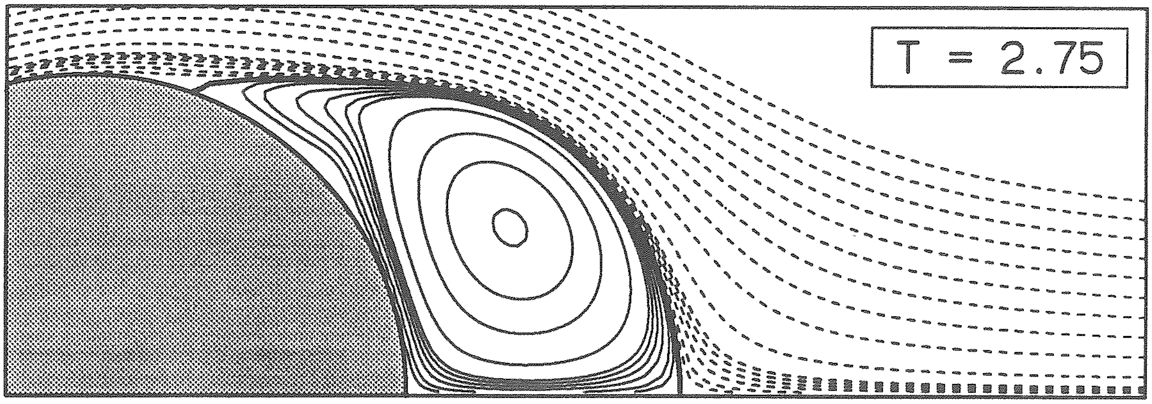

T = 1.50

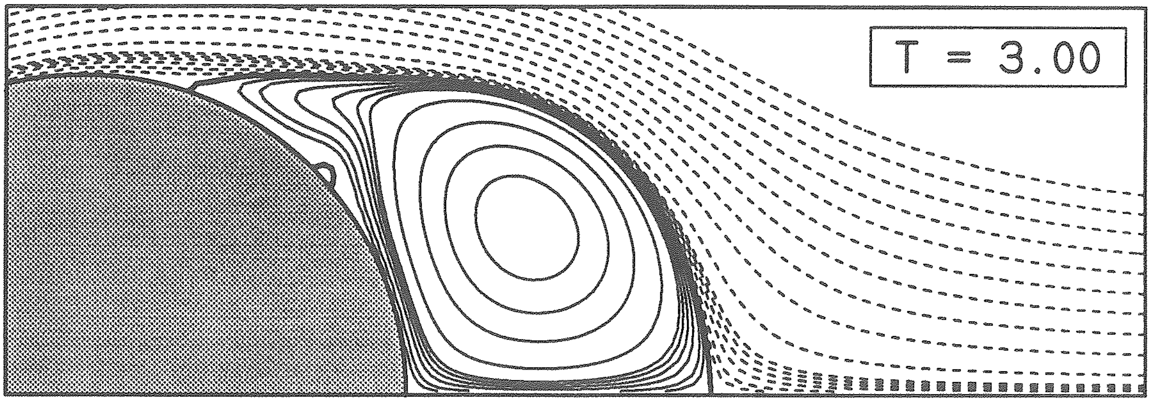

T = 1.75

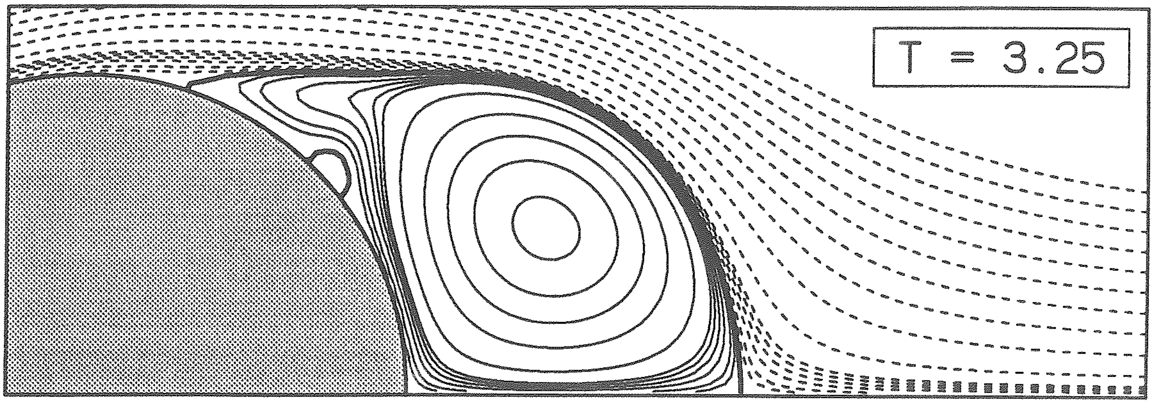

T = 2.00

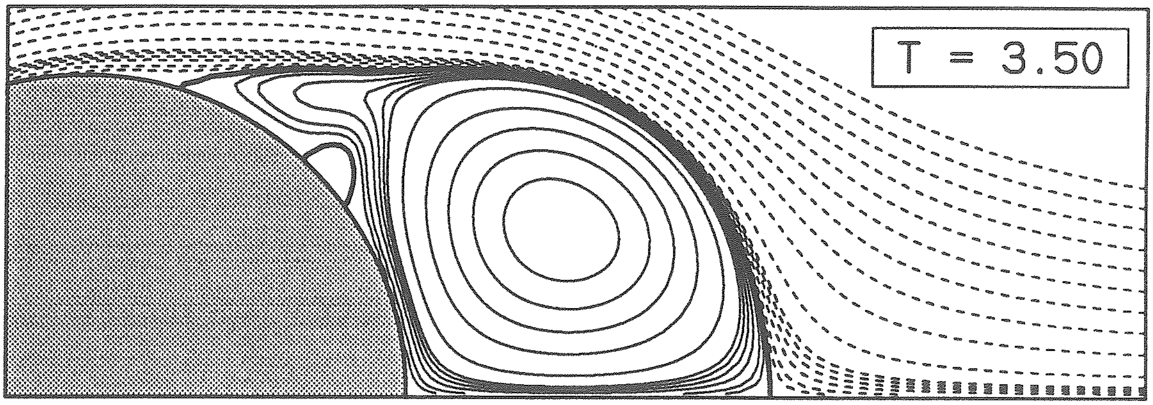

T = 2.25

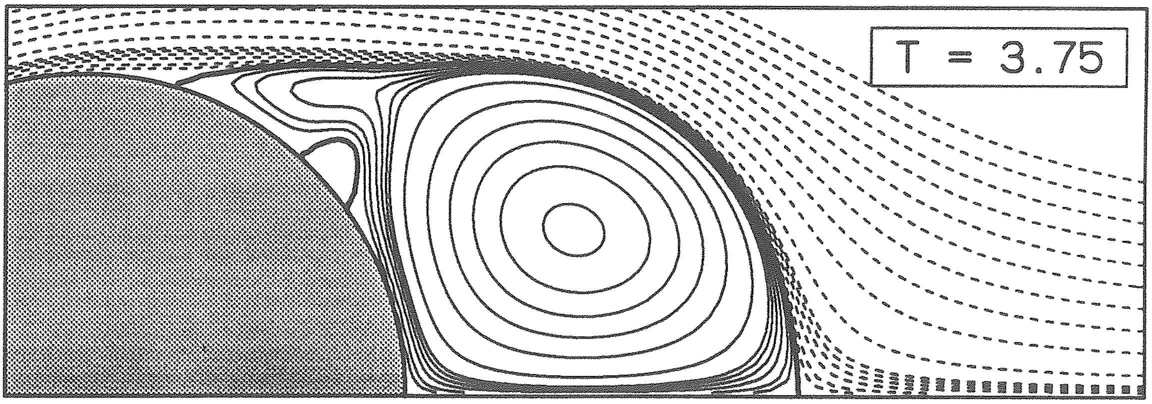

T = 2.50

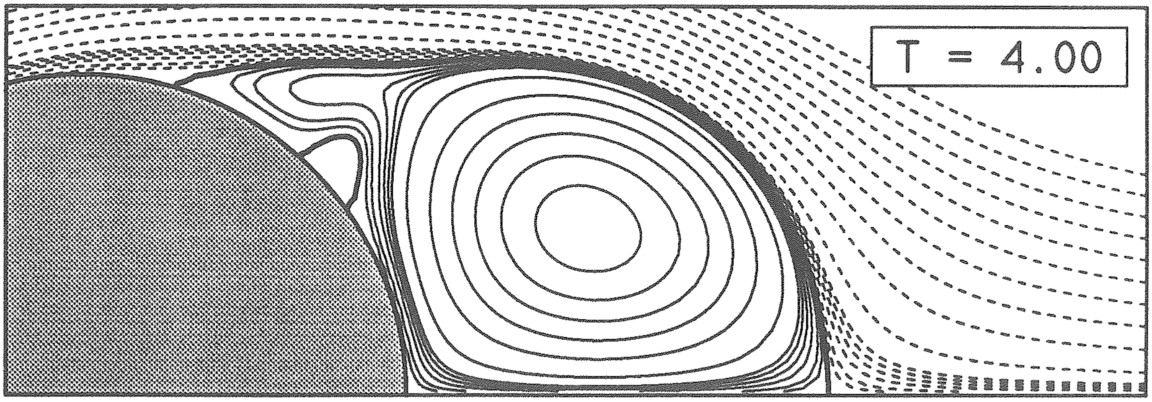

T = 2.75

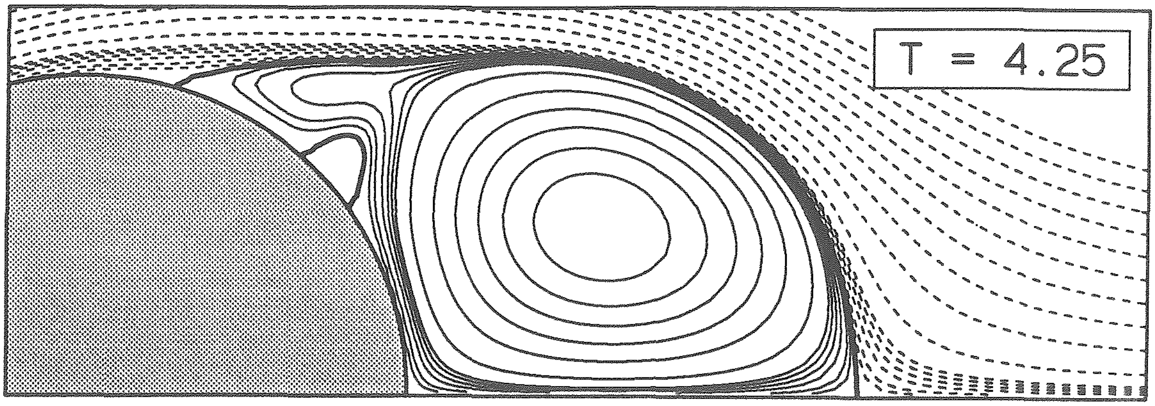

T = 3.00

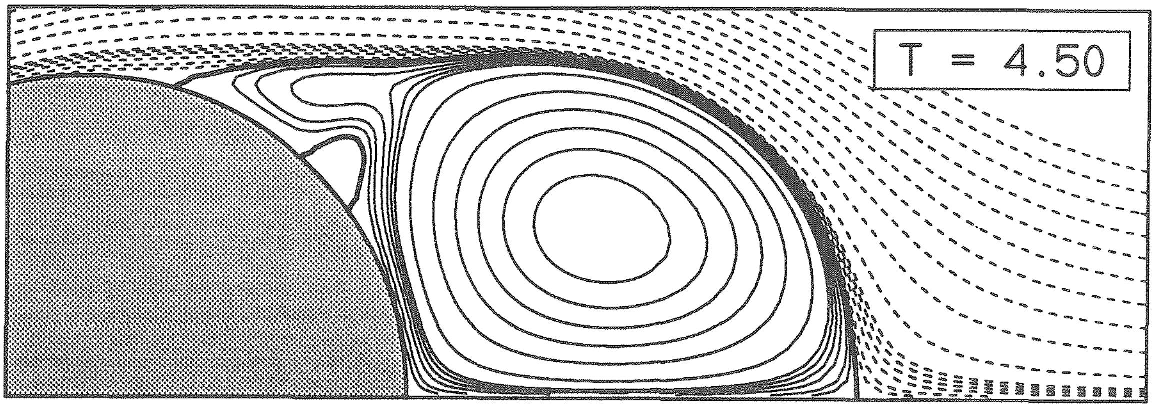

T = 3.25

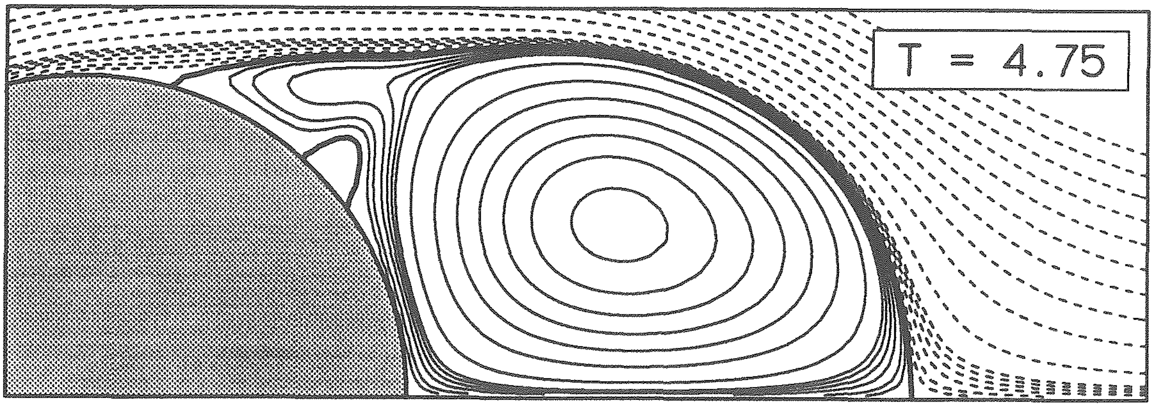

T = 3.50

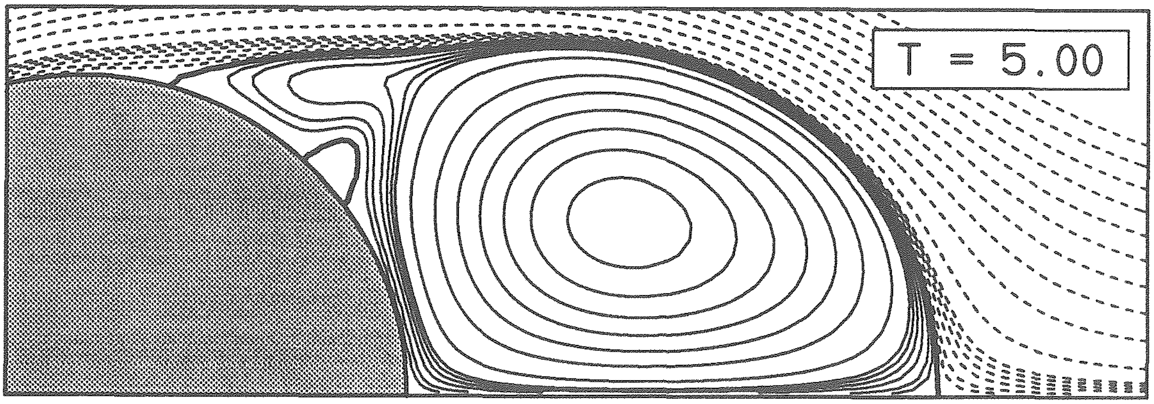

T = 3.75

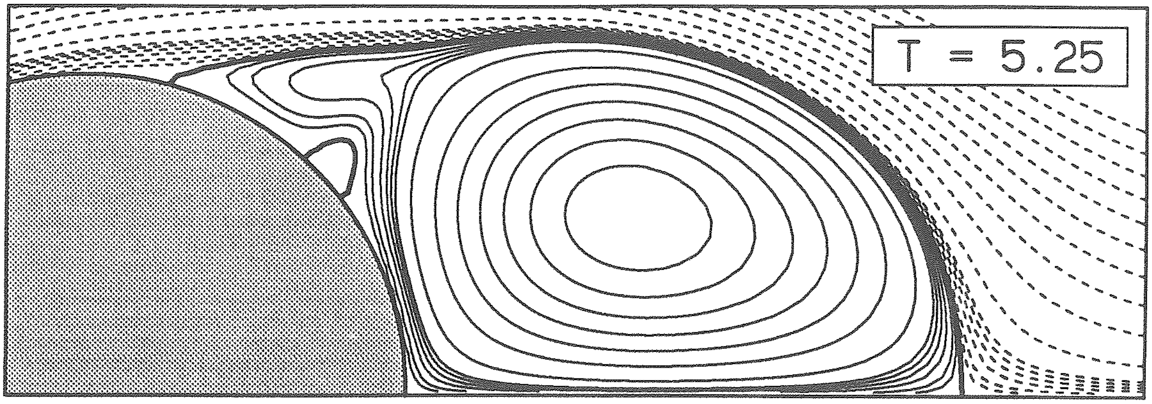

T = 4.00

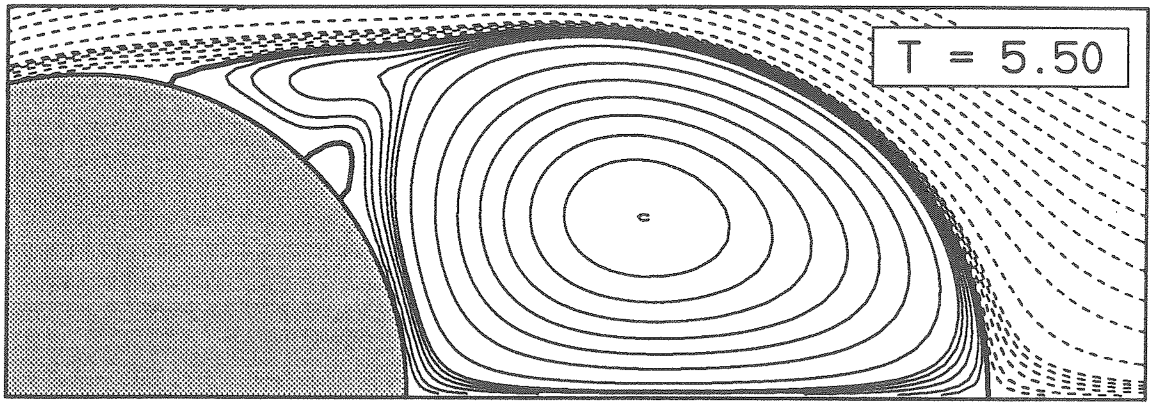

T = 4.25

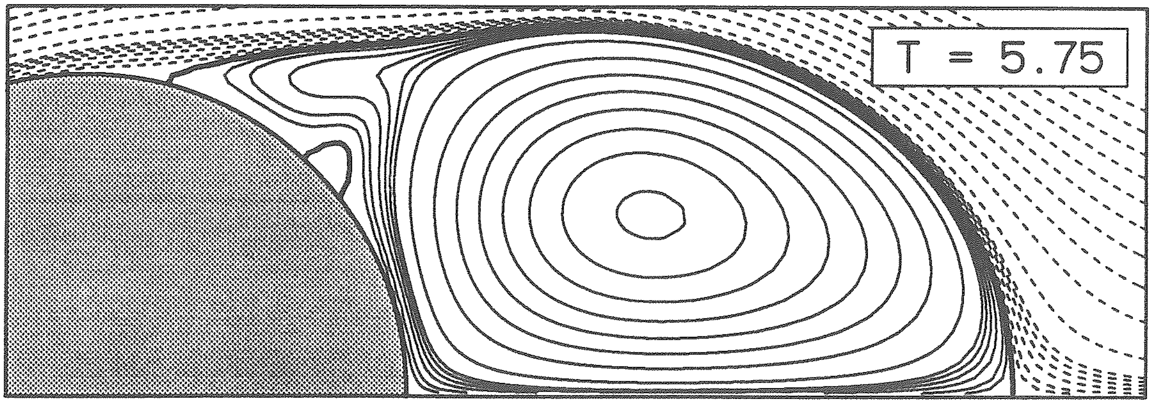

T = 4.50

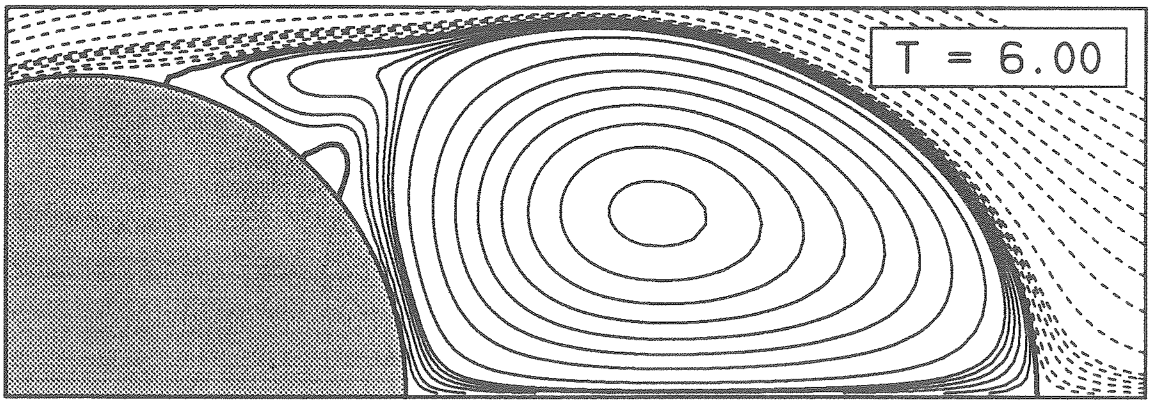

T = 4.75


T = 5.00

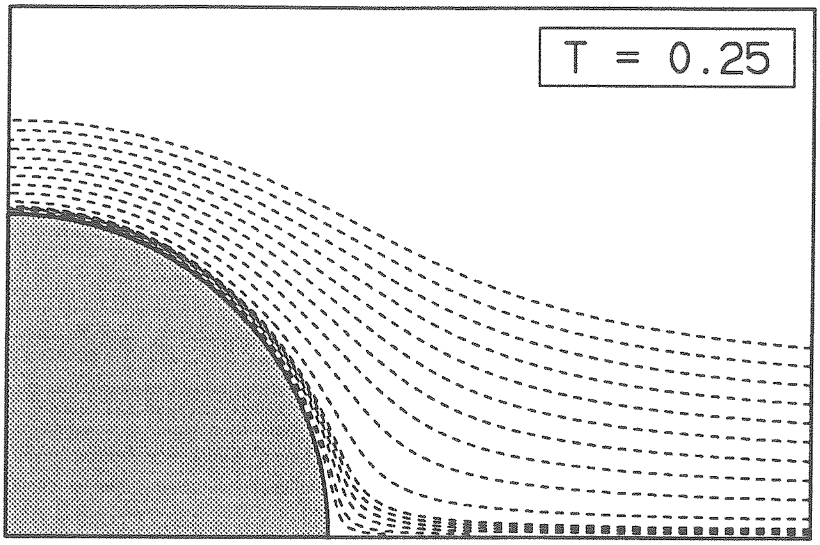

T = 5.25

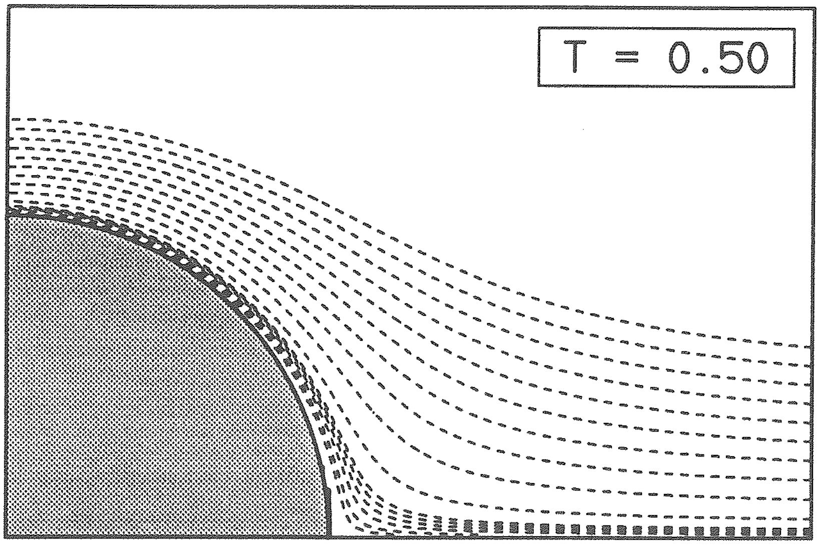

T = 5.50

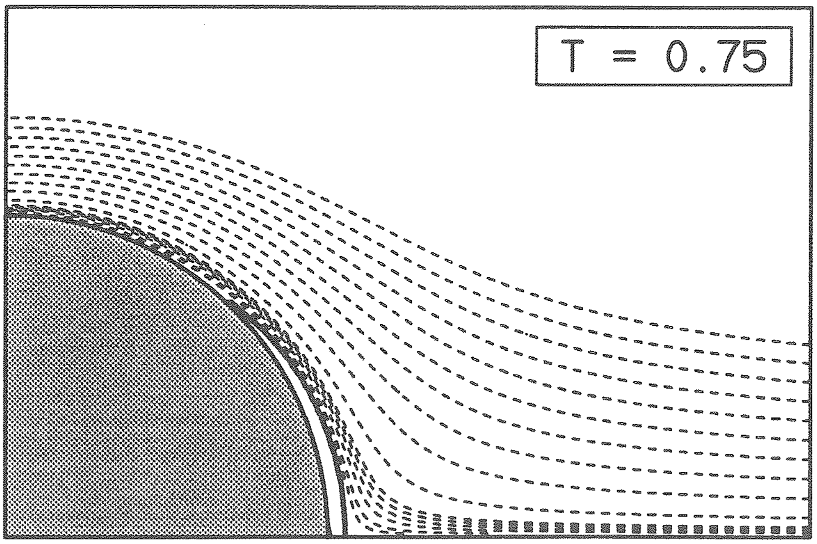

T = 5.75

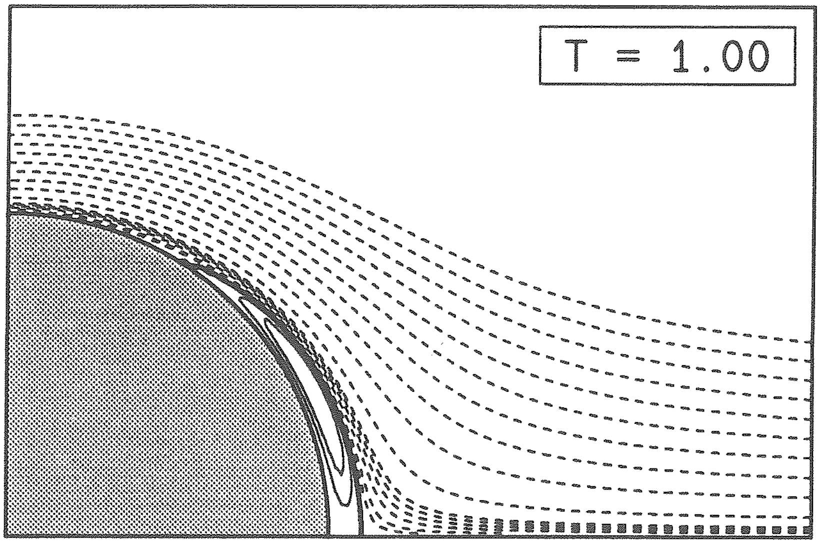

T = 6.00

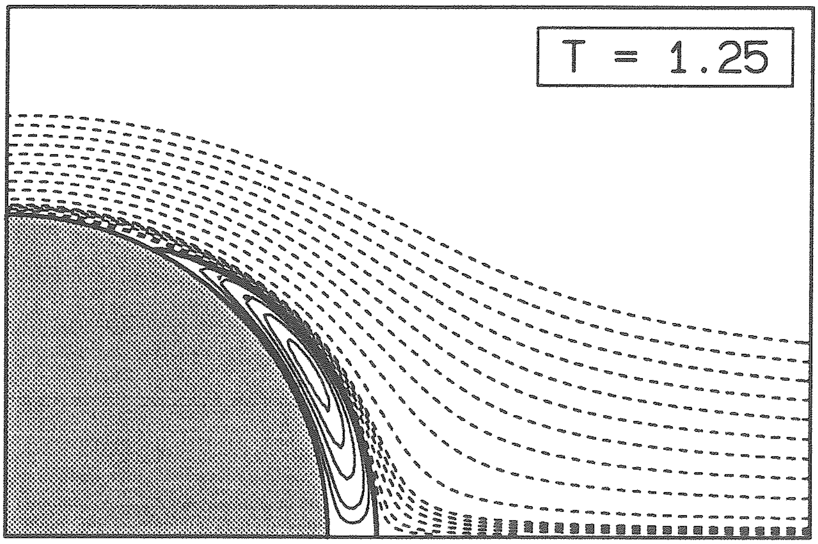
Re = 3000

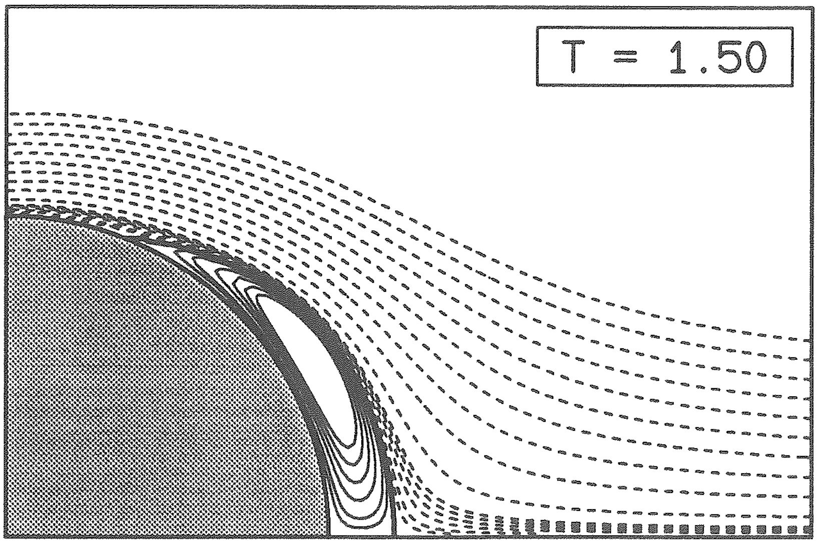

T = 0.25

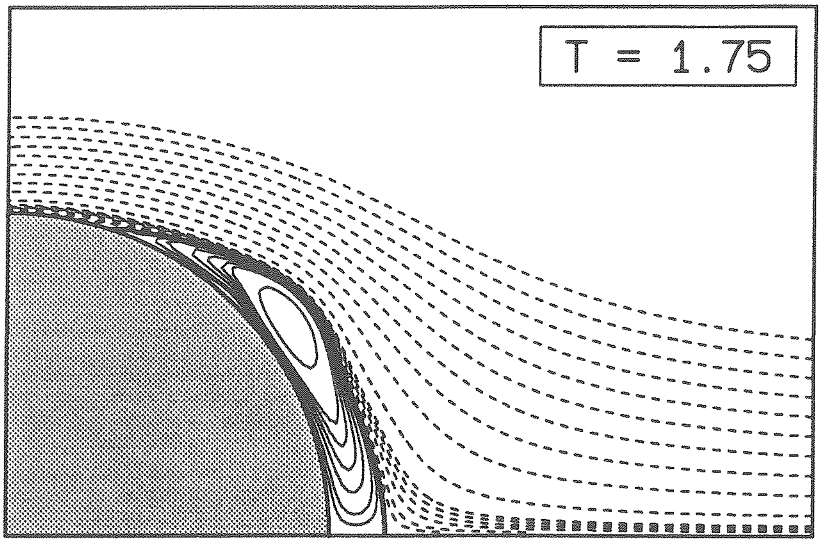

T = 0.50

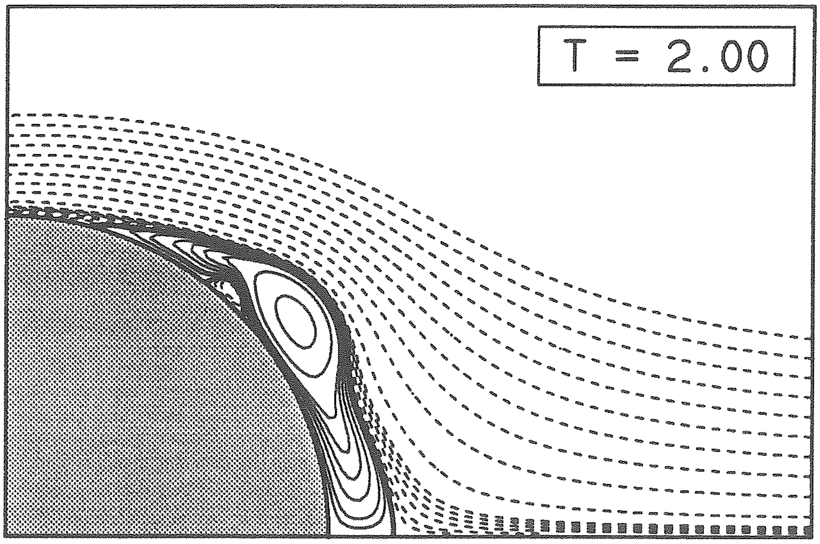

T = 0.75

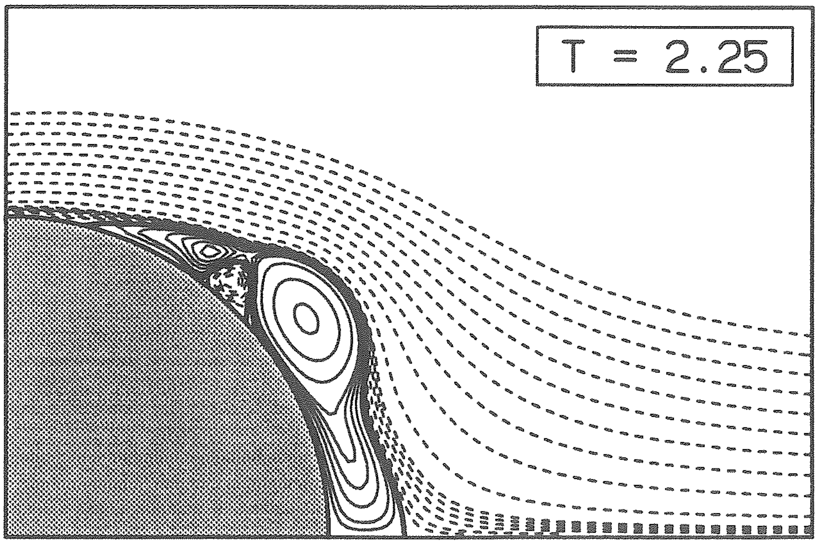

T = 1.00

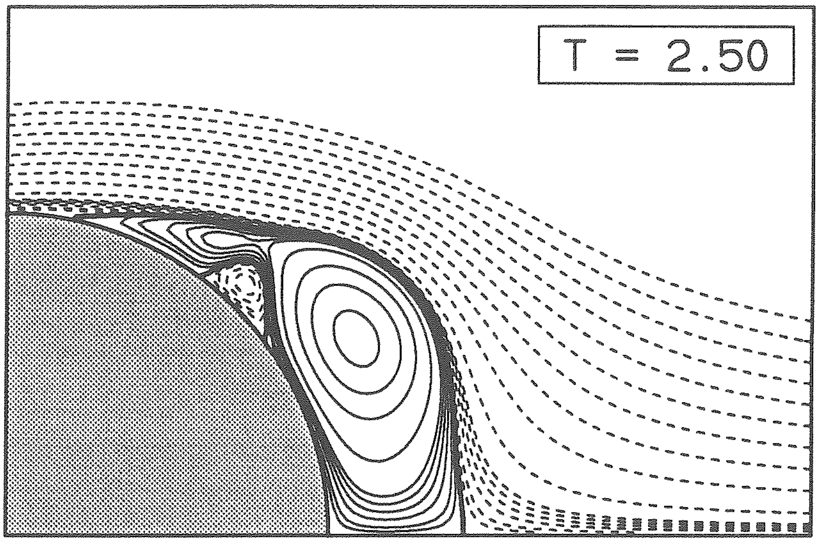

T = 1.25

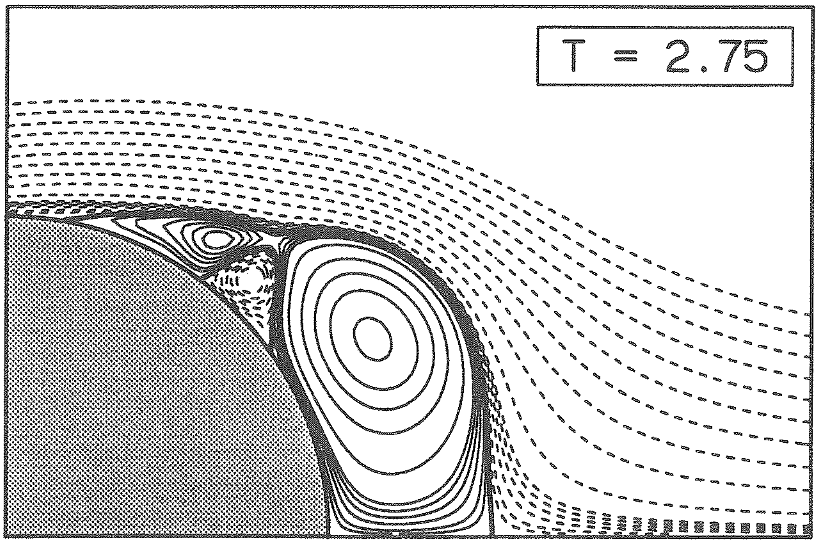

T = 1.50

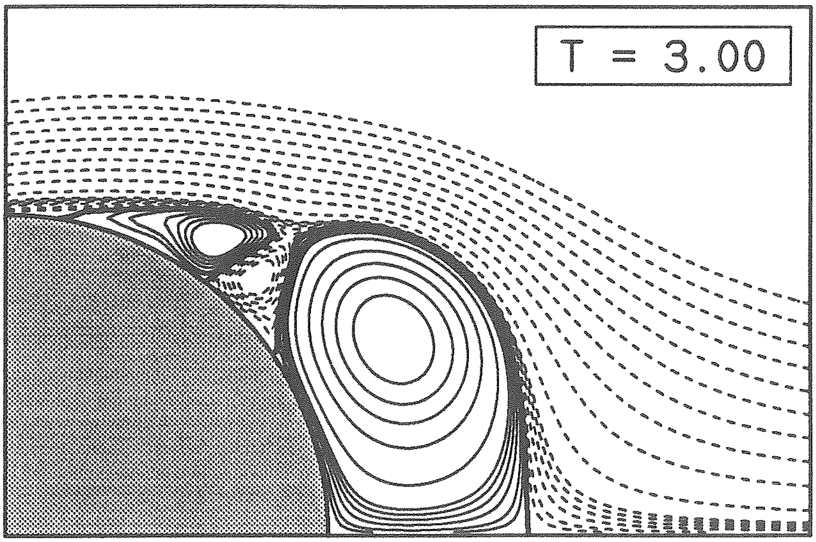

T = 1.75

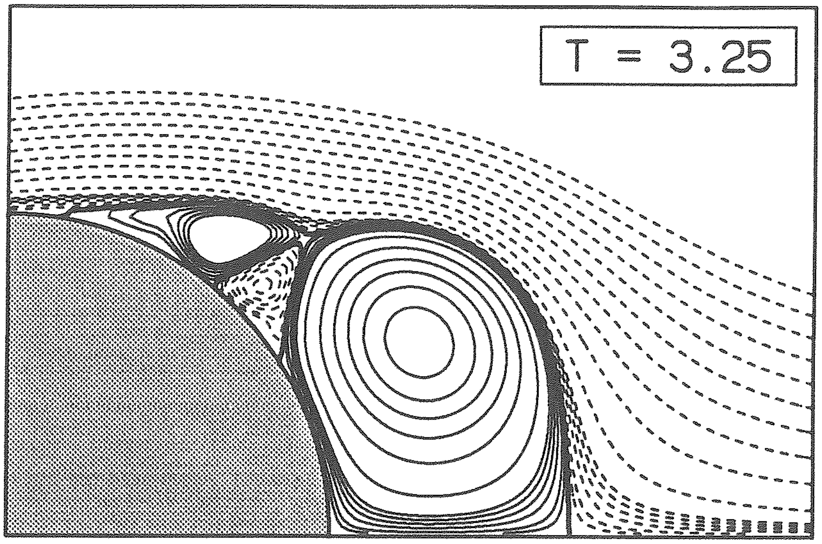

T = 2.00

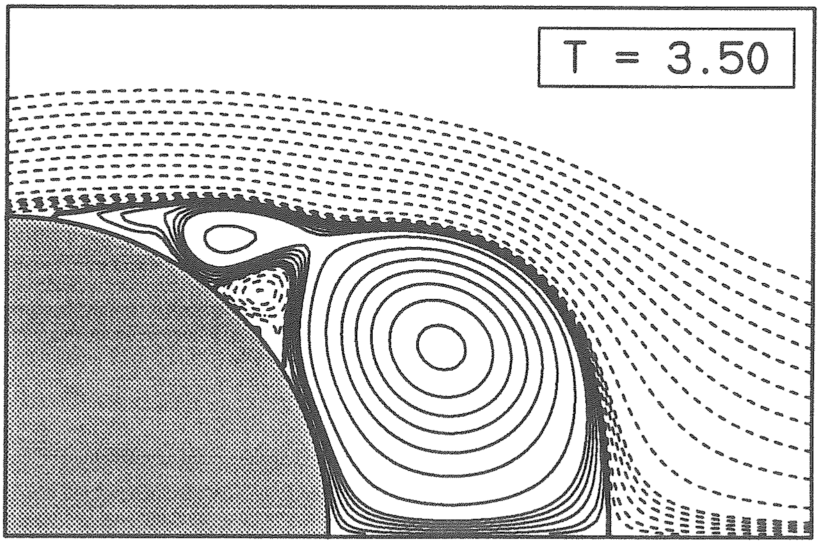

T = 2.25

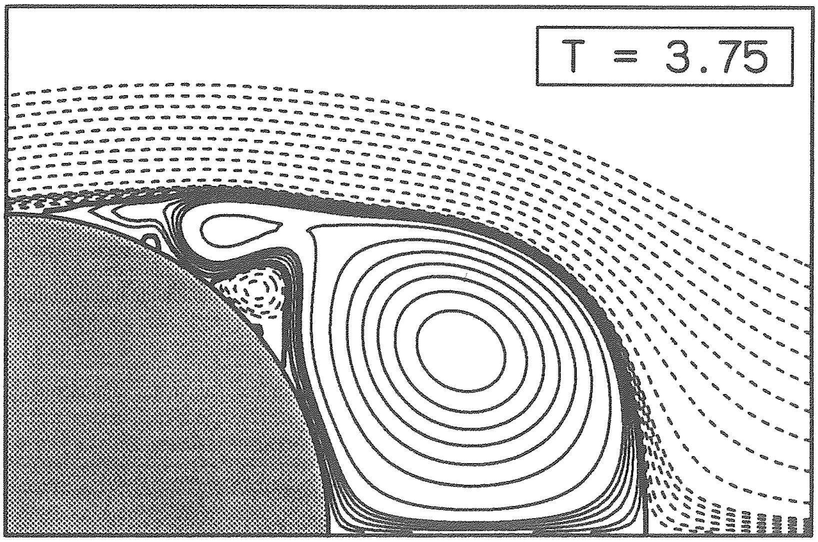

T = 2.50

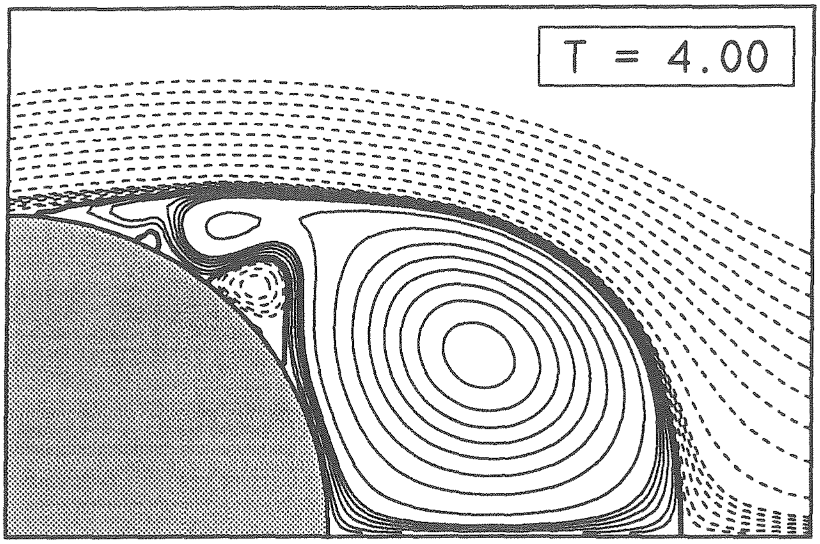

T = 2.75

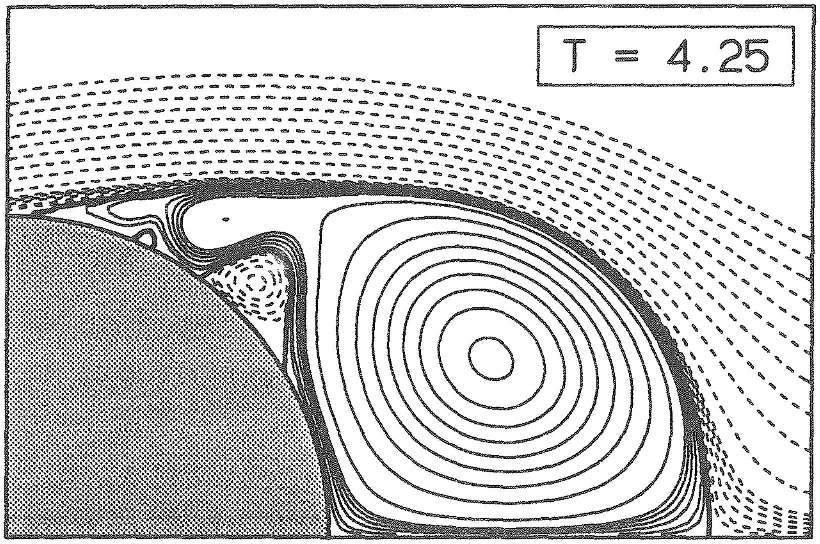

T = 3.00

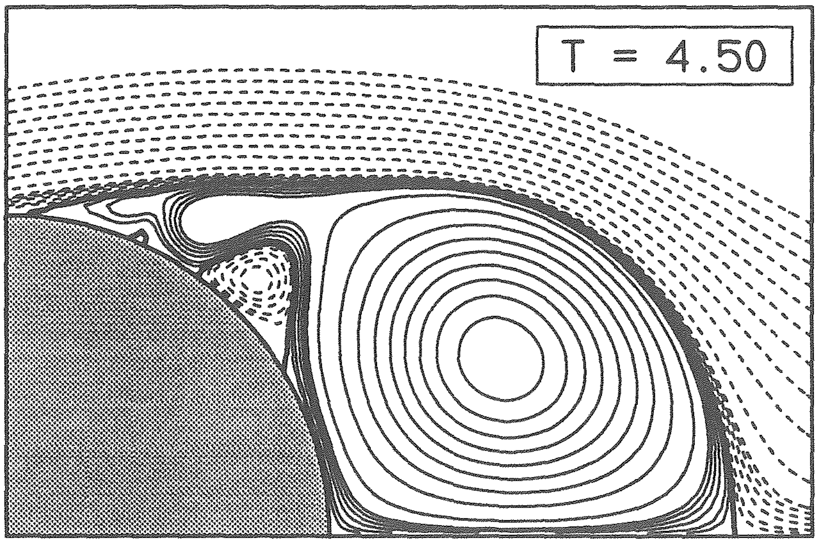

T = 3.25

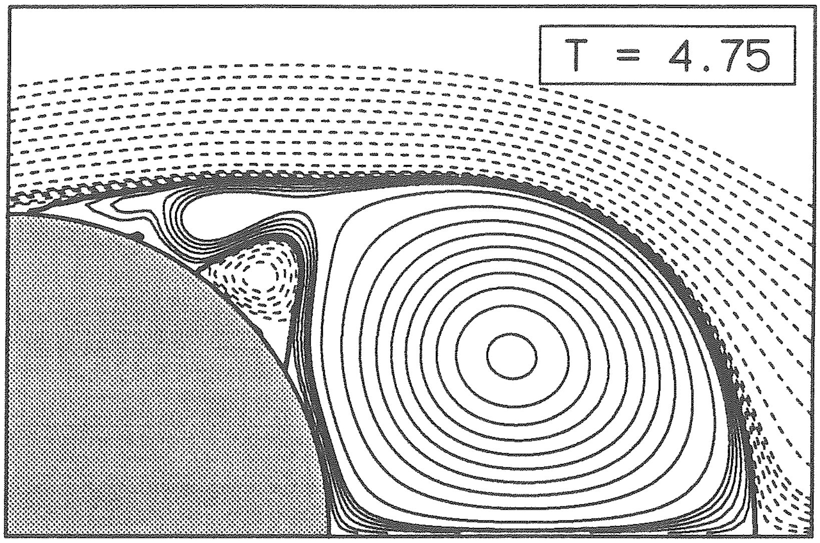

T = 3.50

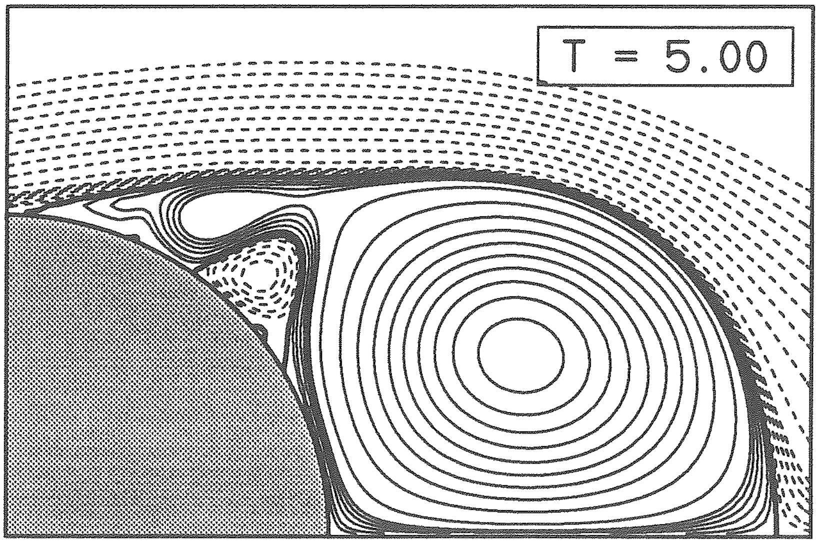

T = 3.75


T = 4.00

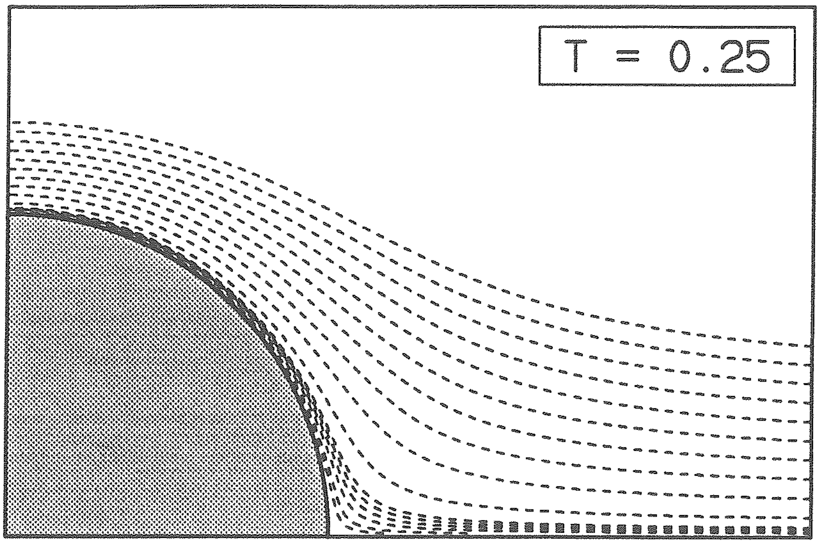

T = 4.25

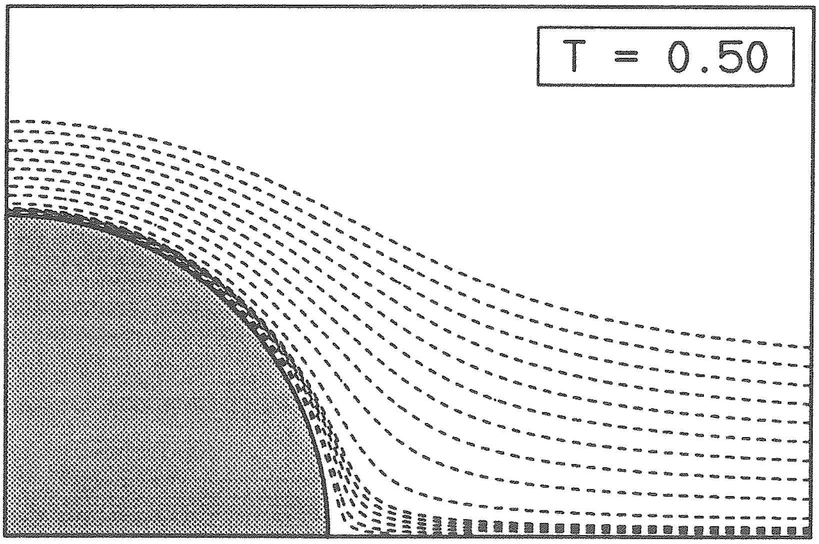

T = 4.50

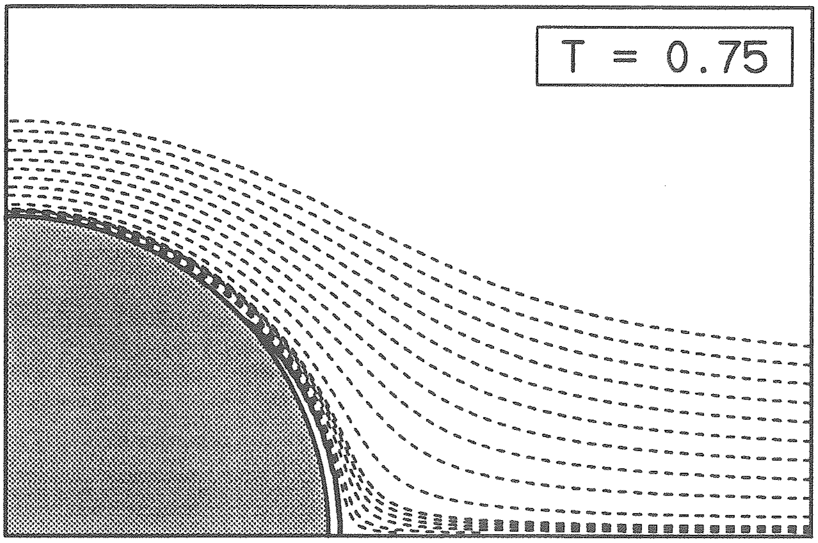

T = 4.75

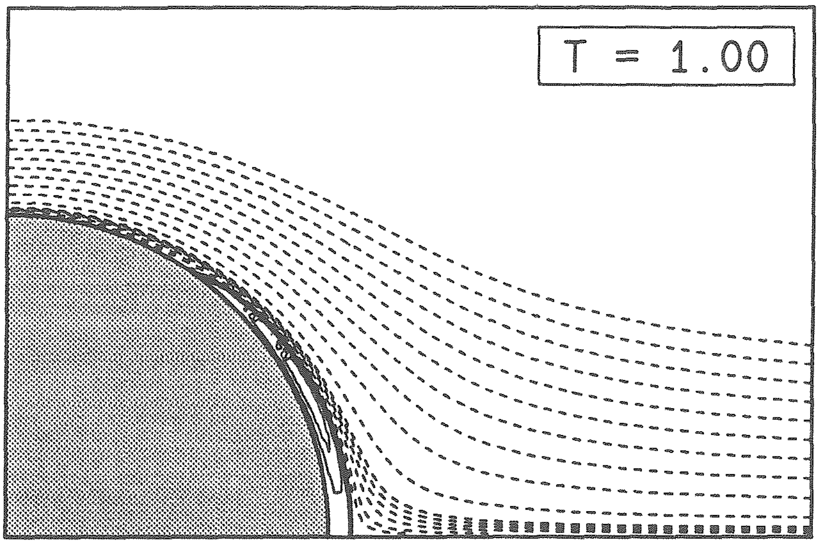

T = 5.00

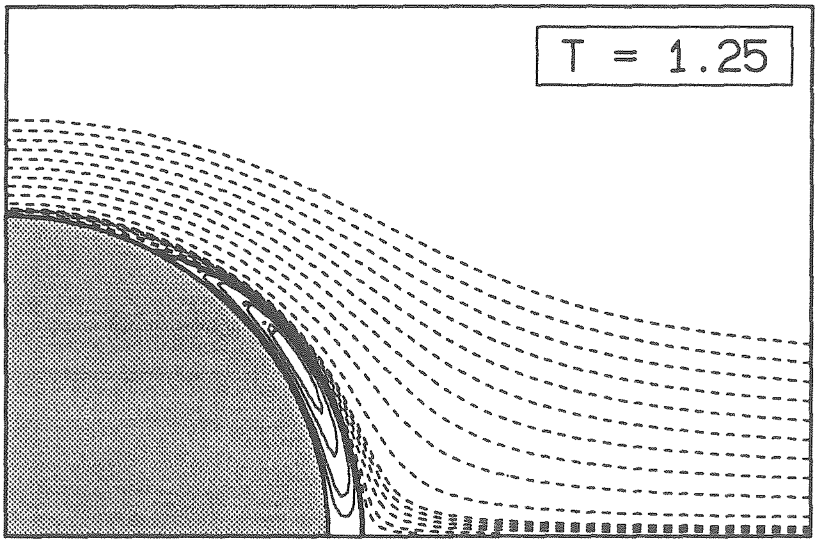
Re = 9500

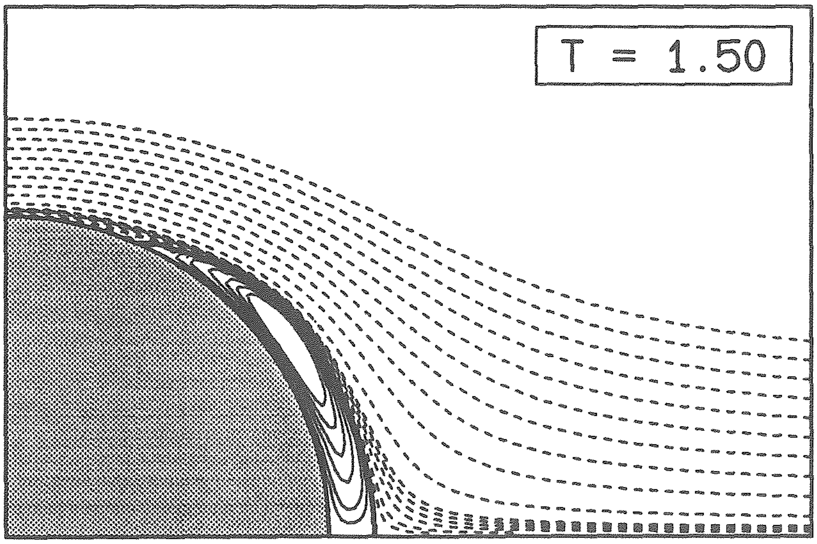

T = 0.25

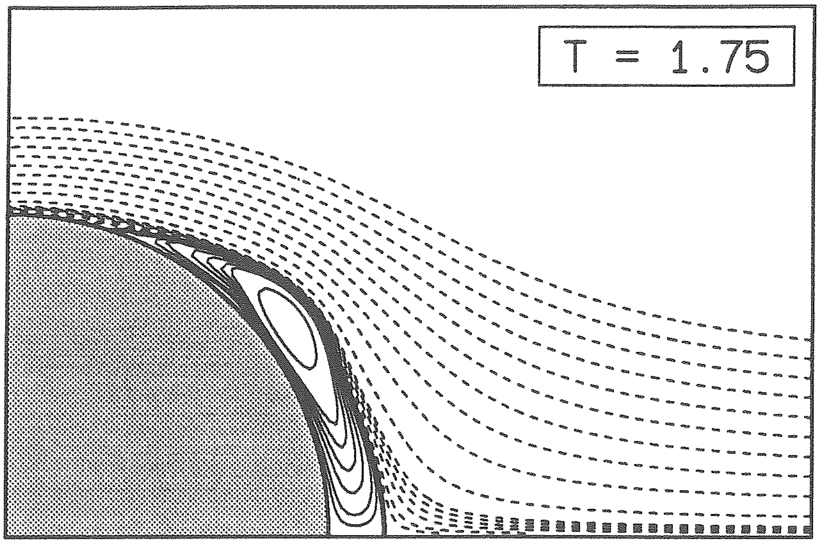

T = 0.50

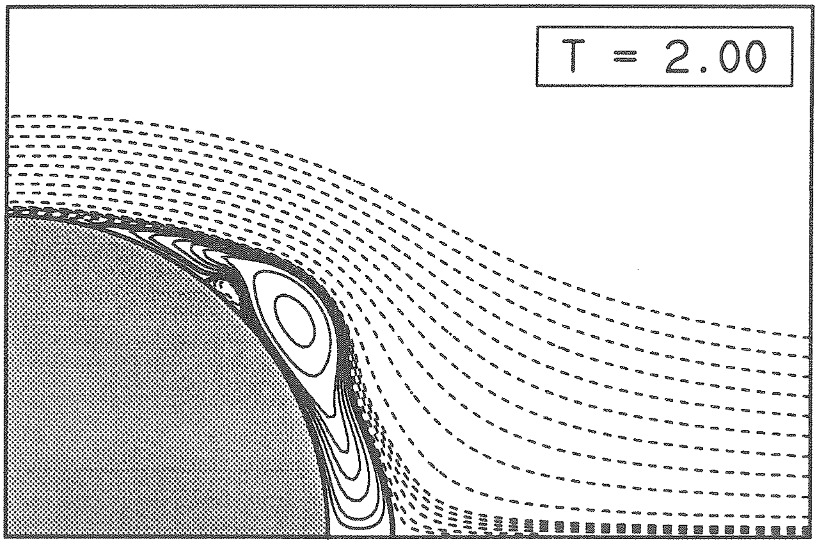

T = 0.75

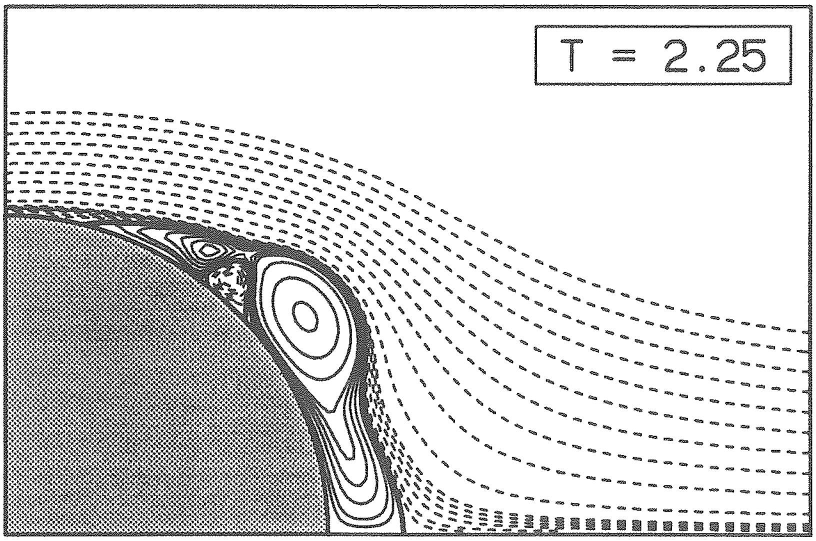

T = 1.00

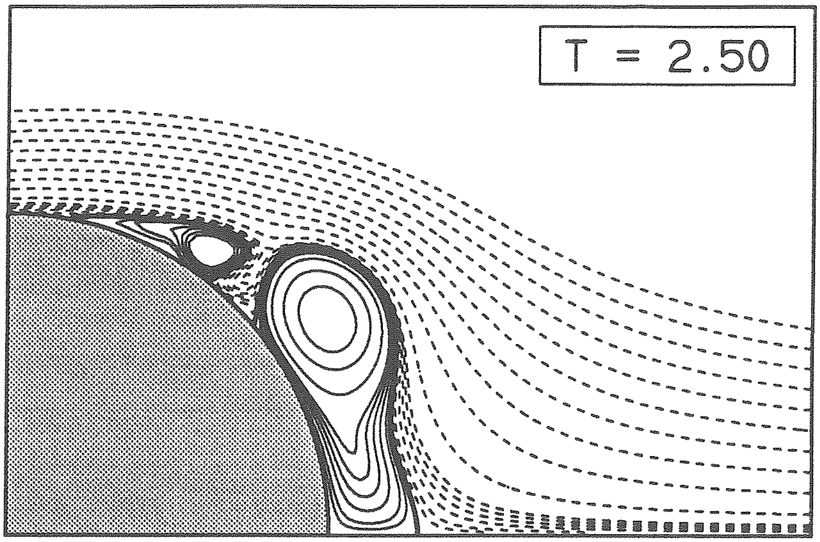

T = 1.25

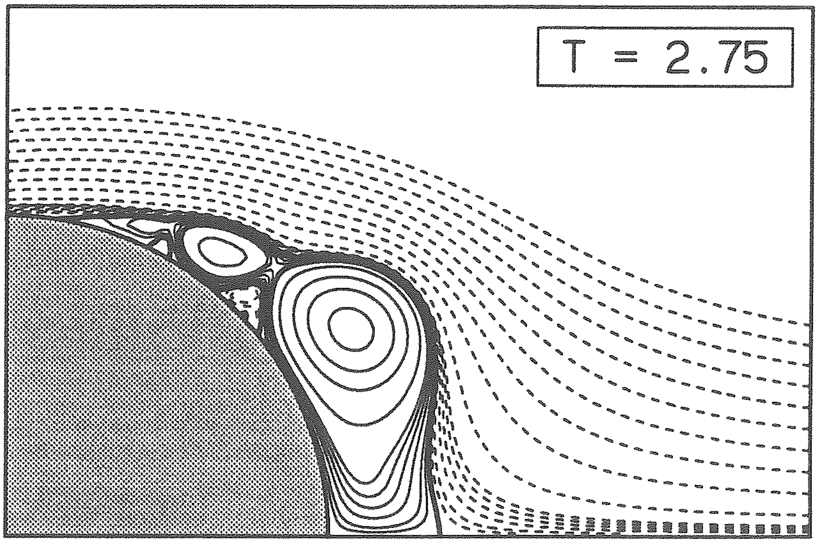

T = 1.50

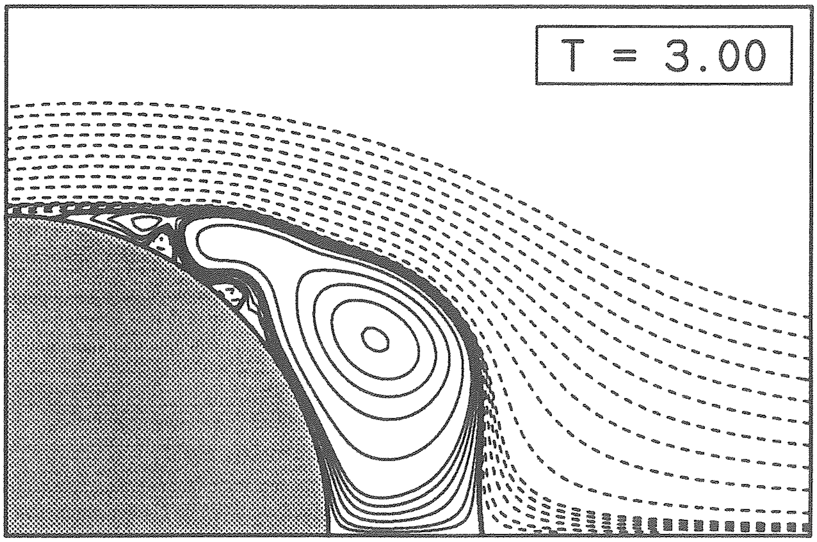

T = 1.75

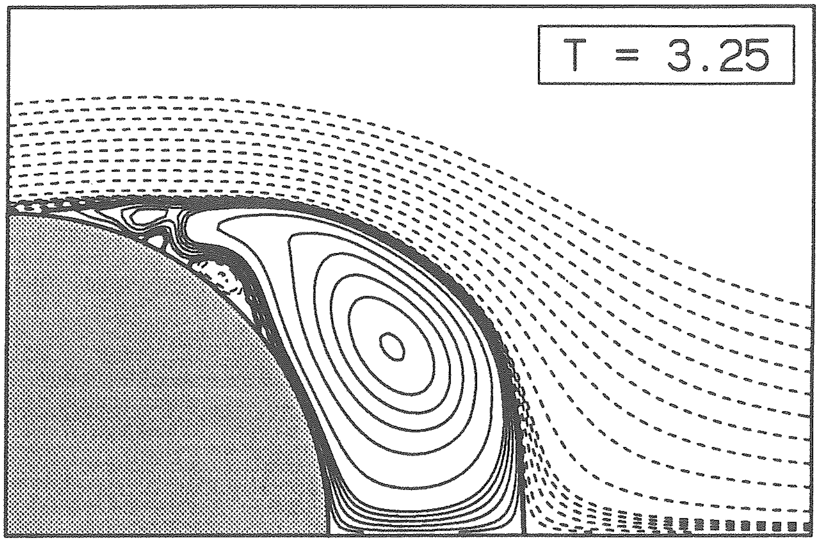

T = 2.00


T = 2.25


T = 2.50


T = 2.75


T = 3.00


T = 3.25

APPENDIX C

Vorticity fields

The next two pages show selected vorticity distributions visualized on an Iris graphic workstation. The vorticity field is reconstructed from its discrete representation and evaluated at the center of every pixel. A color map is used to determine the color assigned to each pixel from the strength of the vorticity at that point. The map is a combination of a continuous background, in blue for positive vorticity and in red for negative values, and a staircase function in green. The discontinuous breaks produced by the staircase function enhances the image and can be thought of as isovorticity contours. The region where the absolute value of the vorticity is small is uniformly shown in green.

The same color map is used for all three Reynolds numbers. However, it has to be stretched as the Reynolds number is increased to accommodate higher values of vorticity.

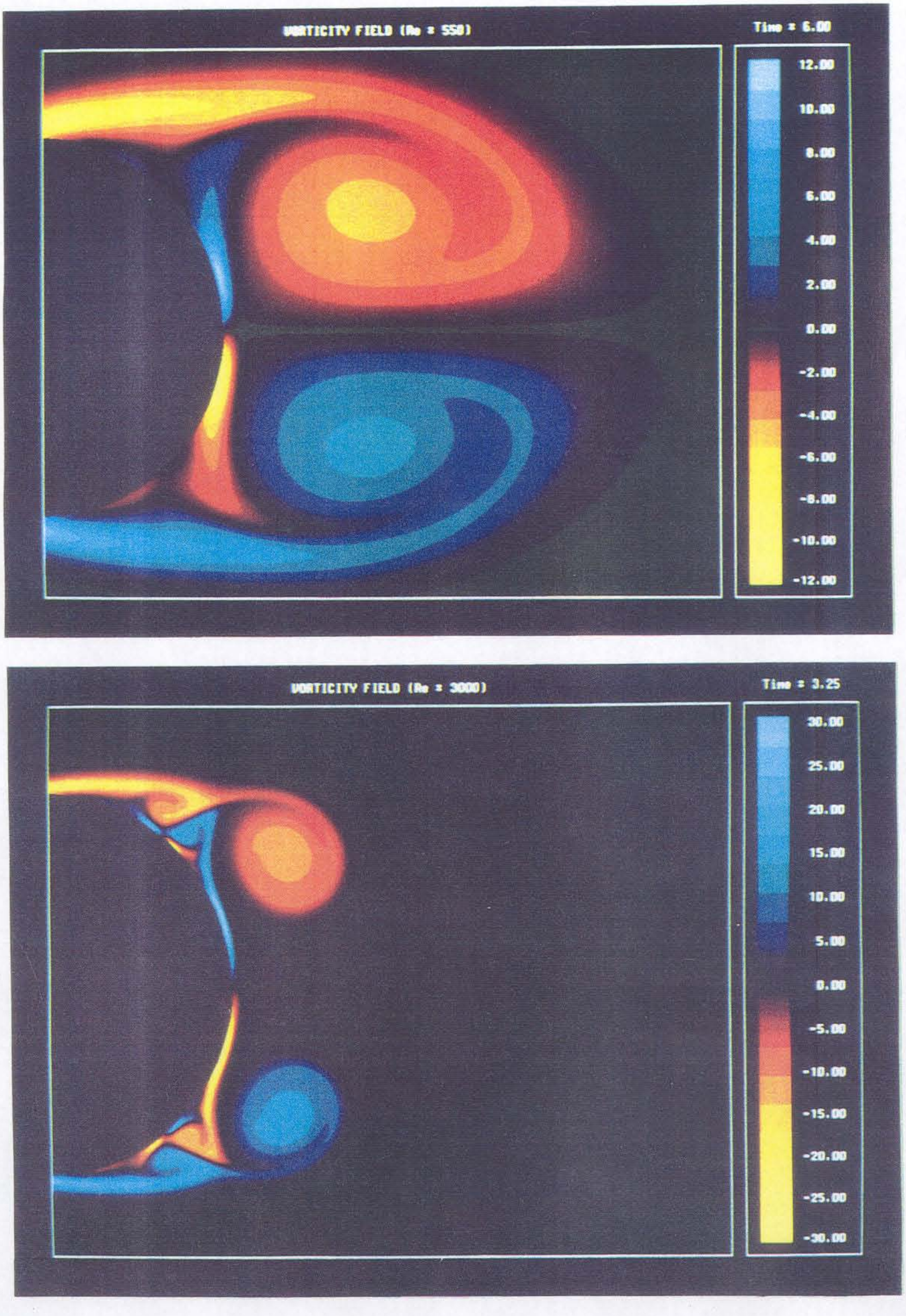


FIGURE C.1 Vorticity field for $Re = 550$ at $T = 6$ (top), and for $Re = 3000$ at $T = 3.25$ (bottom).

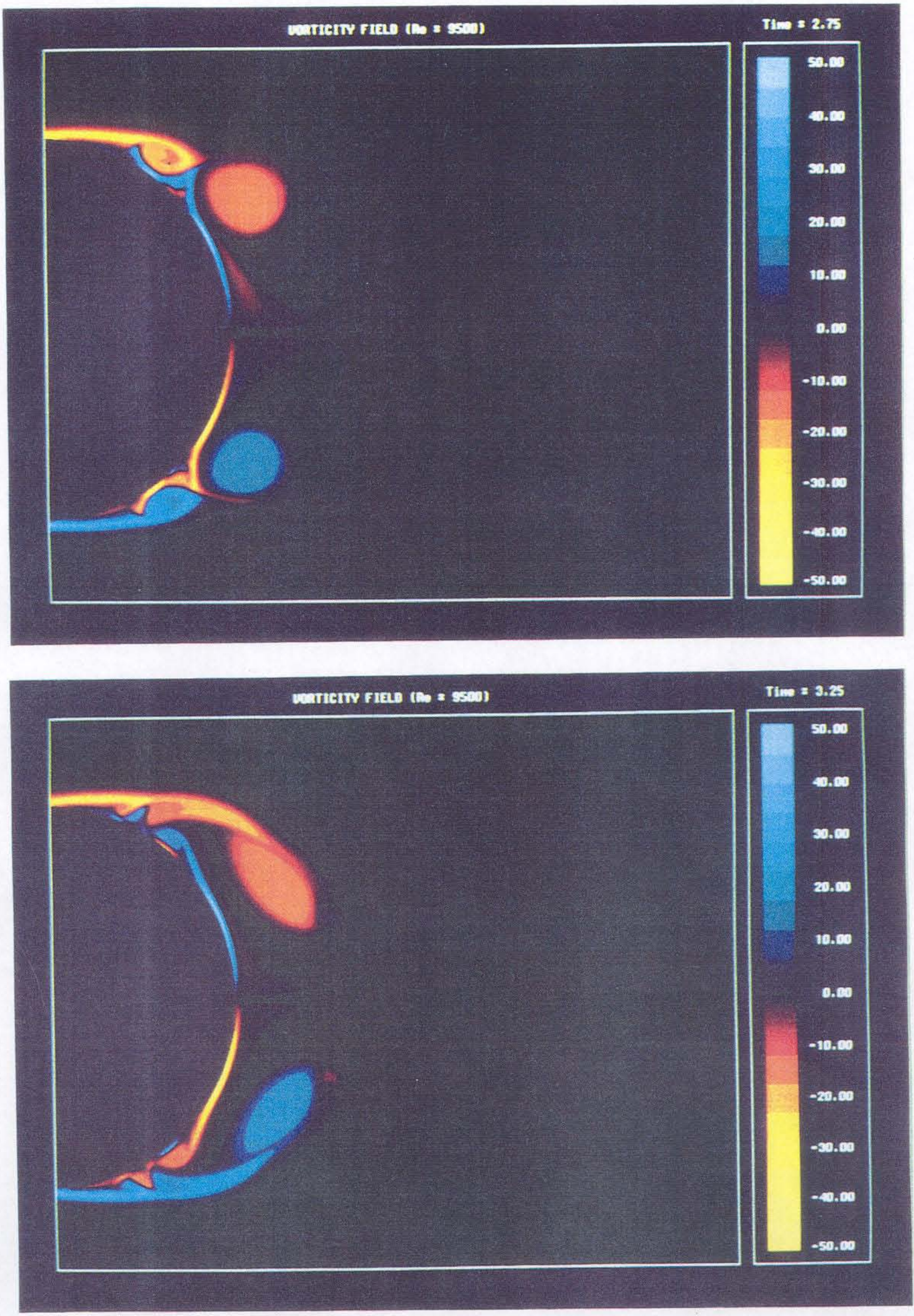


FIGURE C.2 Vorticity field for $Re = 9500$ at $T = 2.75$ (top), and at $T = 3.25$ (bottom).