11 Vortex Methods: An Introduction and Survey of Selected Research Topics

Elbridge Gerry Puckett

11.1 Introduction

Vortex methods are a type of numerical method for approximating the solution of the incompressible Euler or Navier-Stokes equations. In general, vortex methods are characterized by the following three features.

1. The underlying discretization is of the vorticity field, rather than the velocity field. Usually this discretization is Lagrangian in nature and frequently it consists of a collection of particles which carry concentrations of vorticity.
2. An approximate velocity field is recovered from the discretized vorticity field via a formula analogous to the Biot-Savart law in electromagnetism.
3. The vorticity field is then evolved in time according to this velocity field.

In the past two decades a number of different numerical methods for computing the motion of an incompressible fluid have been proposed that have the above features. In this article we consider a class of such methods which are based on the work of Chorin [1973, 1978, 1980, and 1982]. Members of this class are related by the manner in which a vorticity field in an inviscid, incompressible flow is discretized and subsequently evolved. It is common practice to use the term *vortex method* or *the vortex method* to refer to a member of this class when it is used to model the incompressible Euler equations. One can modify the vortex method and use it to model the incompressible Navier-Stokes equations by adding a random walk. This is known as *the random vortex method.* One can also replace the random walk by some non-random technique for solving the diffusion equation. Such methods are generally referred to as *deterministic vortex methods.*

The vortex method can be regarded as a discretization of the equations of fluid motion in vorticity form rather than in the usual velocity-pressure form. This has the advantage of eliminating the pressure from the number of dependent variables to be computed. In addition, as a consequence of the "Biot-Savart Law," the velocity can be written as a linear superposition of basic velocity functions, plus possible corrections for boundary conditions. These velocity functions arise as the convolution of an integral kernel with a concentrated mass of vorticity, usually referred to as a *vortex* or *vortex blob.* This representation of the velocity field as a linear combination of identical velocity functions is both mathematically appealing and relatively straightforward to program. Furthermore, the nature of this discretization is such that diffusive errors due to the numerical method are small. This facilitates relatively inexpensive computations of inviscid or slightly viscous flows – especially when the vorticity is concentrated in small regions of the computational domain.

One of the goals of this article is to provide a reader who is familiar with computational fluid dynamics but not necessarily vortex methods a clear and fairly comprehensive introduction to the subject. We have included a short review of the mathematical theory of vortex methods, especially as it relates to cutoff functions and the influence that these functions have on the convergence rate of the method. We have also included (without proof) the statements of several theorems that have been proved regarding the accuracy and rate of convergence of various vortex method approximations to Euler and Navier-Stokes flow.

In addition, we have endeavored to describe several currently active areas of research in the field. This includes recent work on vorticity boundary conditions and numerical methods for implementing these boundary conditions. We also discuss deterministic vortex methods and describe several of these methods in detail. We briefly mention the recent work on point vortex methods, especially concerning the convergence of the point vortex method to solutions of the incompressible Euler equations in two and three dimensions. Finally, we describe several techniques for reducing the cost of a vortex method computation from $O(N^2)$ to $O(N \log N)$ or even $O(N)$ where N is the number of basic velocity functions in the computation. In particular, we describe the two dimensional versions of the *method of local corrections* and the *fast multipole method* in some detail. It is our hope that we have presented this material in sufficient detail that an interested reader would be able to program many of these methods from our description of them.

This article is *not* a survey of all vortex methods. It is intended to be a short introduction to the field with a brief discussion of selected research topics of interest to the author. Consequently, there has been no attempt to refer to all researchers in the field or all "important work." In fact, there are many aspects of the field which we have not discussed at all. For example, there is no discussion of the vast body of literature concerning the vortex method approximation of a vortex sheet (e.g., see Bernard and Chorin [1973] or Krasny [1986a, 1986b and 1987]) or of the technique of contour dynamics (e.g., Zabusky, Hughes, and Roberts [1979] or Zabusky and Overman [1983]) and related ideas (e.g., Buttke [1990]). On the other hand, we have made a concerted effort to provide as complete a bibliography as possible in those areas that are discussed.

We apologize in advance for any omission or oversight that we have made in this regard. For a more thorough review of vortex methods and more extensive bibliographies the interested reader is referred to the excellent review articles of Beale and Majda [1984], Hald [1991], Leonard [1981 and 1985], and Sethian [1991]. The author has benefited tremendously from reading these articles.

In the following section we review the equations of incompressible fluid flow and introduce various formulations of these equations that will be useful in the subsequent discussion. In Section 11.3 we describe two and three dimensional versions of the vortex method for approximating solutions of the incompressible Euler equations in unbounded regions Ω of $\mathbb{R}^n$. This allows us to ignore the issue of satisfying the no-flow boundary condition on $\partial\Omega$. We discuss the modifications necessary in order to do so in Section 11.4. In Section 11.5 we introduce the random vortex method for approximating solutions of the Navier-Stokes equations in unbounded regions Ω of $\mathbb{R}^n$. We also discuss current research into alternatives to the random walk; i.e., deterministic vortex methods. If Ω has a solid boundary, then one must have a mechanism for approximately satisfying the no-slip boundary condition on $\partial\Omega$. In Section 11.6 we describe a technique for doing this called the *vortex sheet method.* We also mention some recent work on alternatives to this method. Section 11.6 also contains a short discussion on parameter selection for a hybrid random vortex / vortex sheet method computation. At the present this is probably the most widely used type of vortex method for computing viscous flows with boundaries in "engineering" applications. Finally, in Section 11.7 we end with a discussion of "fast vortex methods." This includes a detailed description of two techniques that have been introduced in the past decade: the method of local corrections and the fast multipole method.

The author is indebted to a number of individuals for freely discussing their work and making suggestions during the writing of this article. He would especially like to thank Bill Ashurst, Chris Anderson, Tom Beale, Alexandre Chorin, Claude Greengard, Leslie Greengard, Ole Hald, and Steve Roberts for reading portions of the manuscript.

11.2 The Equations of Incompressible Fluid Flow in Vorticity Form

In this section we derive the Euler and Navier-Stokes equations in vorticity form from the usual velocity-pressure formulation of the equations. We also discuss the issue of boundary conditions, especially recent work on vorticity boundary conditions for the Navier-Stokes equations. We end with a brief introduction to the notion of fractional step methods. A more thorough account of the role that vorticity plays in the mathematical theory of fluid flow may be found in Majda [1986].

11.2.1 The Euler Equations

First we consider the incompressible Euler equations. Let $\Omega \subset I\!R^n$ with $n = 2$ or 3. The incompressible Euler equations are

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p\,, \tag{11.2.1a}$$

$$\nabla \cdot \mathbf{u} = 0\,, \tag{11.2.1b}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}^0(\mathbf{x})\,, \tag{11.2.1c}$$

$$\mathbf{u} \cdot \boldsymbol{\eta} = 0 \qquad \text{on} \qquad \partial\Omega\,, \tag{11.2.1d}$$

where $\mathbf{u}$ is the velocity, p is the pressure, and $\boldsymbol{\eta}$ is a unit vector normal to $\partial\Omega$.[1] We let $\mathbf{x} = (x, y)^T$, $\mathbf{u} = (u, v)^T$ when $n = 2$ and $\mathbf{x} = (x, y, z)^T$, $\mathbf{u} = (u, v, w)^T$ when $n = 3$. Note that equation (11.2.1a) is a vector equation with n components. Equations (11.2.1a–11.2.1d) describe the flow of an incompressible, inviscid fluid in Ω with initial data $\mathbf{u}^0$. We assume that the flow has constant density $\rho = 1$. The boundary condition (11.2.1d) is commonly called the *no-flow boundary condition.*

The Free-Space Problem During the subsequent discussion we will have occasion to consider the case when $\Omega = I\!R^n$. In this event it is understood that the boundary condition (11.2.1d) has been replaced by

$$\mathbf{u}(\mathbf{x}) \to \mathbf{0} \quad \text{as} \quad \mathbf{x} \to \infty\,. \tag{11.2.2}$$

This is sometimes called the *free-space problem* and (11.2.2) is often referred to as the *free-space boundary condition.*

The vorticity $\boldsymbol{\omega}$ is defined by

$$\boldsymbol{\omega} = \nabla \times \mathbf{u}\,. \tag{11.2.3}$$

In two dimensions the vorticity vector $\boldsymbol{\omega} = (0,\, 0,\, \partial_x v - \partial_y u)^T$ points in the direction perpendicular to the (x, y) plane and is usually considered a scalar, $\omega = \partial_x v - \partial_y u$.

Suppose $\Omega = I\!R^n$ and consider the following problem. Given the vorticity $\boldsymbol{\omega}(\mathbf{x}, t)$ at time t, reconstruct the velocity $\mathbf{u}(\mathbf{x}, t)$ such that $\mathbf{u}$ satisfies (11.2.1b), (11.2.2) and (11.2.3). We can do so by finding a function $\boldsymbol{\psi}$ such that

$$\mathbf{u} = \nabla \times \boldsymbol{\psi}\,. \tag{11.2.4}$$

This function can be found by solving

$$\Delta\boldsymbol{\psi} = -\boldsymbol{\omega}\,, \tag{11.2.5a}$$

$$\nabla\boldsymbol{\psi}(\mathbf{x}) \to \mathbf{0} \quad \text{as} \quad \mathbf{x} \to \infty\,. \tag{11.2.5b}$$

[1] We denote vector quantities in bold face type.

Note that $\mathbf{u}$ automatically satisfies (11.2.1b) since $\nabla \cdot (\nabla \times \mathbf{f}) = 0$ for any function $\mathbf{f}$. The function $\boldsymbol{\psi}$ is called the *stream function*. As with the vorticity, in two dimensions $\boldsymbol{\psi}$ points in the direction perpendicular to the (x, y) plane and is generally thought of as a scalar, ψ.

Let G denote the Green's function for the Poisson equation (11.2.5a) in $I\!R^n$. Then we have

$$\boldsymbol{\psi}(\mathbf{x}, t) = \int_{I\!R^n} G(\mathbf{x} - \mathbf{x}')\, \boldsymbol{\omega}(\mathbf{x}', t)\, d\mathbf{x}' = (G * \boldsymbol{\omega})(\mathbf{x}, t)\,, \tag{11.2.6}$$

where $*$ denotes convolution. Since $\partial_x(G * \boldsymbol{\omega}) = (\partial_x G) * \boldsymbol{\omega}$ and similarly for $\partial_y(G * \boldsymbol{\omega})$ and $\partial_z(G * \boldsymbol{\omega})$ we find from (11.2.4) and (11.2.6) that

$$\mathbf{u}(\mathbf{x}, t) = \int_{I\!R^n} (\nabla \times G)(\mathbf{x} - \mathbf{x}')\, \boldsymbol{\omega}(\mathbf{x}', t)\, d\mathbf{x}' = \int_{I\!R^n} \mathbf{K}(\mathbf{x} - \mathbf{x}')\, \boldsymbol{\omega}(\mathbf{x}', t)\, d\mathbf{x}' = (\mathbf{K} * \boldsymbol{\omega})(\mathbf{x}, t) \tag{11.2.7}$$

where $\mathbf{K} = \nabla \times G$. Equation (11.2.7) is frequently referred to as the "Biot-Savart law" in analogy with the well known formula of that name from electrostatics. In two dimensions $\mathbf{K}$ is a vector,

$$\mathbf{K}(\mathbf{x}) = \frac{1}{2\pi|\mathbf{x}|^2}(-y, x)^T\,, \tag{11.2.8}$$

which has a singularity of order $O(|\mathbf{x}|^{-1})$ at $\mathbf{x} = \mathbf{0}$. In three dimensions $\mathbf{K}$ is a matrix,

$$\mathbf{K}(\mathbf{x}) = \frac{1}{4\pi|\mathbf{x}|^3}\begin{pmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{pmatrix}, \tag{11.2.9}$$

with a singularity of order $O(|\mathbf{x}|^{-2})$ at $\mathbf{x} = \mathbf{0}$. The singularity in $\mathbf{K}$ has played a central role in the mathematical theory of vortex methods.

Note that if $\boldsymbol{\omega}$ has compact support or decays sufficiently fast as $\mathbf{x} \to \infty$, then $\mathbf{u}$ given by (11.2.7) automatically satisfies the free-space boundary condition (11.2.2). Thus, when $\Omega = I\!R^n$, we have found a way to write the velocity $\mathbf{u}$ in terms of the vorticity $\boldsymbol{\omega}$.

The No-Flow Boundary Condition When $\Omega \subset I\!R^n$ but $\Omega \neq I\!R^n$ the function $\mathbf{u}$ given by (11.2.7) will not necessarily satisfy the no-flow boundary condition (11.2.1d) on $\partial\Omega$. We can rectify this situation by writing the velocity $\mathbf{u}$ as the sum of a "rotational" flow $\mathbf{u}_\omega$ and an irrotational (vorticity free) flow $\mathbf{u}_p$ as follows. Given the vorticity field $\boldsymbol{\omega}(\mathbf{x})$ in Ω let $\mathbf{u}_\omega = \mathbf{K} * \boldsymbol{\omega}$ (where we define $\boldsymbol{\omega}(\mathbf{x}) = 0$ for $\mathbf{x} \notin \Omega$ for the purposes of the integration over $I\!R^n$ in (11.2.7)). In general, $\mathbf{u}_\omega$ will fail to satisfy (11.2.1d) on $\partial\Omega$. We seek a scalar function ϕ in Ω such that

$$\Delta\phi = 0 \tag{11.2.10a}$$

$$\frac{\partial \phi}{\partial \boldsymbol{\eta}} = -\mathbf{u}_\omega \cdot \boldsymbol{\eta} \qquad \text{on} \quad \partial\Omega \,. \tag{11.2.10b}$$

In other words, we seek a solution to the Laplace equation (11.2.10a) in Ω with Neumann boundary conditions (11.2.10b). The theory of elliptic equations (e.g., see Garabedian [1986]) guarantees the existence of a solution ϕ under reasonable restrictions on $\partial\Omega$ and the boundary data.

The *potential flow* $\mathbf{u}_p$ defined by

$$\mathbf{u}_p(\mathbf{x}) = \nabla\phi(\mathbf{x}) \,, \tag{11.2.11}$$

satisfies (11.2.1b) since $\nabla \cdot \mathbf{u}_p = \Delta\phi = 0$. Also, $\nabla\times\mathbf{u}_p = 0$ since $\nabla\times\nabla f = 0$ for any scalar function f. Now consider the velocity field given by,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_\omega(\mathbf{x}) + \mathbf{u}_p(\mathbf{x}) \,. \tag{11.2.12}$$

It is apparent from the foregoing that $\mathbf{u}$ satisfies $\nabla\cdot\mathbf{u} = 0$ and $\nabla\times\mathbf{u} = \boldsymbol{\omega}$. Furthermore, it follows from (11.2.10b), (11.2.11) and (11.2.12) that

$$\mathbf{u}\cdot\boldsymbol{\eta} = \mathbf{u}_\omega\cdot\boldsymbol{\eta} - \mathbf{u}_\omega\cdot\boldsymbol{\eta} = 0 \qquad \text{on} \quad \partial\Omega \,.$$

Hence $\mathbf{u} = \mathbf{u}_\omega + \mathbf{u}_p$ satisfies (11.2.1b,d) and (11.2.3) as desired.

In a vortex method computation we construct an approximation $\tilde{\mathbf{u}}$ to the exact solution $\mathbf{u}$ of (11.2.1a–d) in precisely this manner: Given an approximation $\tilde{\boldsymbol{\omega}}$ to the vorticity field $\boldsymbol{\omega}$ we take $\tilde{\mathbf{u}}_\omega = \mathbf{K} * \tilde{\boldsymbol{\omega}}$ to be our approximation to $\mathbf{u}_\omega$. Then, in order to satisfy the no-flow boundary condition (11.2.1d), we add to $\tilde{\mathbf{u}}_\omega$ an approximation $\tilde{\mathbf{u}}_p$ to the potential flow $\mathbf{u}_p$ by solving the Neumann problem (11.2.10a–b) with boundary data $-\tilde{\mathbf{u}}_\omega\cdot\boldsymbol{\eta}$. In Section 11.4 we will discuss various techniques for computing $\tilde{\mathbf{u}}_p$.

11.2.2 The Euler Equations in Vorticity Form

By taking the curl of equations (11.2.1a,c) we obtain the vorticity form of the Euler equations,

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} \,, \tag{11.2.13a}$$

$$\nabla\cdot\mathbf{u} = 0 \,, \tag{11.2.13b}$$

$$\boldsymbol{\omega} = \nabla\times\mathbf{u} \,, \tag{11.2.13c}$$

$$\boldsymbol{\omega}(\mathbf{x},0) = \boldsymbol{\omega}^0(\mathbf{x}) \,, \tag{11.2.13d}$$

$$\mathbf{u}(\mathbf{x},t)\cdot\boldsymbol{\eta}(\mathbf{x}) = \mathbf{0} \qquad \text{for} \qquad \mathbf{x}\in\partial\Omega \,. \tag{11.2.13e}$$

Here $D/Dt \equiv \partial_t + (u \cdot \nabla)$ is the so-called *material derivative*. Note that by taking the curl of (11.2.1a) we have eliminated the pressure from the set of equations to be solved. This greatly simplifies the task of finding approximations to $\mathbf{u}$. Furthermore, from (11.2.7), (11.2.10a–b) and (11.2.12) we see that the velocity depends linearly on the vorticity. The vortex method takes advantage of this fact by discretizing the vorticity field as a linear superposition of patches of vorticity. Each vortex patch induces a velocity field via (11.2.7) that is easy to compute, and the total velocity field is simply the sum of these velocities plus a correction for boundary conditions.

11.2.3 The Navier-Stokes Equations

We now turn to the Navier-Stokes equations. Let ν denote the kinematic viscosity. In vorticity form the Navier-Stokes equations are given by,

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu\Delta\boldsymbol{\omega}\,, \tag{11.2.14a}$$

$$\nabla \cdot \mathbf{u} = 0\,, \tag{11.2.14b}$$

$$\boldsymbol{\omega} = \nabla \times \mathbf{u}\,, \tag{11.2.14c}$$

$$\boldsymbol{\omega}(\mathbf{x}, 0) = \boldsymbol{\omega}^0(\mathbf{x})\,, \tag{11.2.14d}$$

$$\mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Omega\,. \tag{11.2.14e}$$

These equations describe the flow of a viscous, incompressible fluid in Ω with initial data $\boldsymbol{\omega}^0$. They can be derived from the usual velocity-pressure form of the Navier-Stokes equations in the same way we derived equations (11.2.13a–e) from (11.2.1a–d).

From a computational point of view these equations present two additional difficulties over the Euler equations. The first difficulty is that of accurately modeling the diffusive effects of the term $\nu\Delta\boldsymbol{\omega}$ on the right hand side of (11.2.14a). In particular, for very small ν solutions of (11.2.14a–e) can have important features which vary on spatial scales that are $O(\sqrt{\nu})$. This leads to the requirement that a numerical method must resolve spacial features of the flow that are $O(\sqrt{\nu})$. For example, in a finite difference computation this requirement would lead to the need for $O(\sqrt{\nu^{-1}})$ grid points per spacial dimension (e.g., see Henshaw, Kreiss, and Reyna [1991]). This can become prohibitively expensive as $\nu \to 0$. We will return to this issue in our discussion of the random vortex method in Section 11.5.

The other difficulty that has been introduced is the addition of a second boundary condition

$$\mathbf{u}(\mathbf{x}, t) \cdot \boldsymbol{\tau}(\mathbf{x}) = 0 \quad \text{for} \quad \mathbf{x} \in \partial\Omega\,, \tag{11.2.15}$$

where $\boldsymbol{\tau}$ is a unit vector tangent to $\partial\Omega$. This is usually referred to as the *no-slip boundary condition* (or sometimes, the *stick boundary condition*). This boundary condition has a very important physical interpretation: it accounts for the creation of vorticity at $\partial\Omega$. In fact, it can be shown that in an incompressible flow this is the only means by which vorticity is created (e.g. see Chorin and Marsden [1979], Schlichting [1979], or White [1974]).

It is not immediately apparent how best to approximate the boundary condition (11.2.15) within the context of a vortex method computation. Chorin [1973 and 1978] has devised several methods in which concentrations of vorticity – i.e. vortices – are added at points on $\partial\Omega$ with strengths chosen so that the resulting velocity field approximately cancels $\tilde{\mathbf{u}} \cdot \boldsymbol{\tau}$ on $\partial\Omega$ where $\tilde{\mathbf{u}}$ is a numerical approximation to (11.2.12). These vortices are subsequently allowed to diffuse into Ω and participate in the flow. The process of diffusing the vortices into the flow is accomplished by letting each vortex undergo a random walk. This method has the attraction of mimicking the physical process of vorticity creation at $\partial\Omega$. In one form or another this idea is at the basis of almost all methods currently being used to approximate (11.2.15) in vortex method computations.

There has been much research into clarifying the mathematical principles underlying Chorin's idea – as well as into finding alternatives to it. We briefly discuss some of these efforts here. Let Ω be a compact subset of $I\!R^2$ which lies in the interior of a closed curve $\partial\Omega$. Let $\mathbf{u}$ be an incompressible velocity field in Ω that satisfies the no-flow boundary condition (11.2.1d) on $\partial\Omega$ and let $\omega = \nabla \times \mathbf{u}$ be the corresponding vorticity field. The stream function ψ associated with $\mathbf{u}$ may be found by solving

$$\Delta\psi = -\omega\,, \tag{11.2.16a}$$

$$\psi = 0 \qquad \text{on} \quad \partial\Omega\,. \tag{11.2.16b}$$

Therefore ψ satisfies (11.2.4),

$$\mathbf{u}(\mathbf{x}) = (\psi_y(\mathbf{x}), -\psi_x(\mathbf{x}))\,, \tag{11.2.17}$$

and (11.2.1d),

$$\mathbf{u} \cdot \boldsymbol{\eta} = 0 \qquad \text{on} \qquad \partial\Omega\,.$$

Now suppose that $\mathbf{u}$ also satisfies the no-slip boundary condition (11.2.15) on $\partial\Omega$. Then we must have

$$\frac{\partial\psi}{\partial\boldsymbol{\eta}} = 0 \qquad \text{on} \quad \partial\Omega\,. \tag{11.2.18}$$

This constitutes a second boundary condition on ψ – one in addition to (11.2.16b). In other words, the problem of finding ψ from ω is now over determined.

Quartapelle and Valz-Gris [1981] considered the question "What are the conditions on the vorticity so that the stream function ψ constructed by solving (11.2.16a,b) also satisfies the boundary condition (11.2.18)?" Their answer was that ω must be orthogonal to all harmonic functions η in Ω. In other words, that

$$\int_{\Omega} \omega(\mathbf{x})\,\eta(\mathbf{x})\,d\mathbf{x} = 0\,, \tag{11.2.19}$$

for all η harmonic in Ω. Thus, Quartapelle and Valz-Gris have replaced the boundary condition (11.2.15) by a constraint on the evolution on the vorticity (11.2.19). They implemented this constraint numerically by advancing the solution of (11.2.14a) one time step without regard for this constraint and then projecting the result down onto the component which satisfies (11.2.19). A detailed implementation of this algorithm is presented in Quartapelle [1981].

Anderson [1989] studied the connection between Chorin's [1978] algorithm and the Quartapelle and Valz-Gris algorithm. He was able to show that Quartapelle and Valz-Gris' constraint on the vorticity was equivalent to altering the vorticity at $\partial\Omega$ by exactly the same amount Chorin added at $\partial\Omega$.

It is natural to inquire about the possibility of finding a boundary condition on the vorticity $\boldsymbol{\omega}$ consistent with the boundary condition (11.2.14e) on $\mathbf{u}$. In this regard Cottet [1988a] has shown that for domains $\Omega \subseteq I\!R^2$ the following system of equations are equivalent to the Navier-Stokes equations (11.2.14a–e) (see also Mas-Gallic [1990]),

$$\frac{D\omega}{Dt} = \nu\Delta\omega\,, \tag{11.2.20a}$$

$$\omega(\mathbf{x},0) = \boldsymbol{\omega}^0(\mathbf{x})\,, \tag{11.2.20b}$$

$$\omega = \nabla \times \mathbf{u} \qquad \text{on} \quad \partial\Omega\,, \tag{11.2.20c}$$

$$\Delta\mathbf{u} = -\nabla \times \omega\,, \tag{11.2.20d}$$

$$\mathbf{u} = \mathbf{0} \qquad \text{on} \ \partial\Omega\,. \tag{11.2.20e}$$

One can regard these equations as two coupled subsystems (11.2.20a–c) and (11.2.20d–e). The first system is an evolution equation for the vorticity (11.2.20a) with initial condition (11.2.20b) and boundary condition (11.2.20c). The second system is Poisson's equation (11.2.20d) with Dirichlet boundary data (11.2.20e). Both Cottet [1988a] and Mas-Gallic [1990] have proposed numerical methods based on these equations for solving the Navier-Stokes equations in $I\!R^2$. One of the essential differences between these algorithms and Chorin's algorithm is that the algorithms of Cottet and Mas-Gallic do not use a random walk. Such methods are commonly referred to as *deterministic vortex methods*. We will return to this topic in Section 11.5.

We also mention that Hou and Wetton [1990] have constructed a finite difference method for approximating solutions of the Navier-Stokes equations in vorticity-stream formulation which incorporate a vorticity boundary condition. They have proved the convergence of this method to solutions of the Navier-Stokes equations. This is currently a very active area of research. We refer the interested reader to Anderson [1989], Cottet [1988a], Hou and Wetton [1990], Mas-Gallic [1990], Quartapelle [1981], and Quartapelle and Valz-Gris [1981] for further details and references.

11.2.4 The Diffusion Equation

Solutions of the Navier-Stokes equations (11.2.14a–e) may be approximated by alternately solving two distinct systems of partial differential equations for small time steps Δt; with the solution of the first system being the initial data for the second and so on. The first of these systems is the Euler equations (11.2.13a–e). The second is the *diffusion* or *heat equation,*

$$\omega_t = \nu \Delta \omega \,, \tag{11.2.21a}$$

$$\mathbf{u} = \mathbf{0} \quad \text{on} \quad \partial\Omega \,. \tag{11.2.21b}$$

Note that this is not the usual boundary value problem for the heat equation – the boundary data is prescribed for the velocity **u** rather than the vorticity ω. This is simply a reflection of the fact that we are using the boundary condition (11.2.14e) on the velocity with the evolution equations (11.2.14a–d) for the vorticity. It is natural to inquire if one can replace (11.2.21b) with an equivalent boundary condition on the vorticity. This is essentially the same question that is being addressed in the work on vorticity boundary conditions for equations (11.2.14a–d).

Equation (11.2.21a) describes the rate at which the vorticity is diffusing in the flow. In many instances it is possible to arrange the computation so that if the no-flow boundary condition (11.2.1d) is satisfied by the initial data to (11.2.21a,b), then the solution of (11.2.21a) will satisfy (11.2.1d) at later times as well. Thus, the boundary condition usually associated with the solution of (11.2.21a) is the no-slip boundary condition (11.2.15). This boundary condition may be regarded as a source term that appears during the solution of (11.2.21a) (e.g., see Puckett [1989a], p. 310).

11.2.5 Fractional Step Methods

The idea of solving the Euler equations for a small time step Δt and then the heat equation for the same time step and using the result as an approximation to the solution of the Navier-Stokes equations at time Δt is sometimes called *viscous splitting*. This is a special case of a more general technique known as *operator splitting*. Viscous splitting for the Navier-Stokes equations in $\Omega = I\!R^n$ for $n = 2, 3$ with the free space boundary condition (11.2.2) has been given a rigorous justification by Beale and Majda [1981], Ebin and Marsden [1970], and Ying [1990]. The problem is much more difficult when boundaries are included. Work on this problem appears in Benfatto and Pulverenti [1983] and Alessandrini, Douglis, and Fabes [1983].

Numerical methods based on operator splitting are called *fractional step* methods. Such methods are applicable to a wide variety of partial differential equations. In particular, the random vortex method is a fractional step method for approximating solutions of the Navier-Stokes equations. The first step consists of a vortex method approximation to the solution of the Euler equations (11.2.13a–e) in which the no-flow boundary condition is satisfied by adding a potential flow. The second step is a random walk approximation to the solution of the heat equation (11.2.21a) in which the no-slip boundary condition (11.2.15) is satisfied by creating vortex elements on $\partial\Omega$. The initial data for one step is the result obtained from the previous step.

11.3 The Vortex Method

The vortex method is a numerical method for approximating solutions of the incompressible Euler equations (11.2.1a–d). It is a *particle method* in which fluid particles carrying concentrations of vorticity are followed as their positions and concentrations evolve with the motion of the fluid. Fundamental to all particle methods is the notion of the fluid flow map $\mathbf{x} : \Omega \times [0, T] \rightarrow \Omega$ defined so that $\mathbf{x}(\boldsymbol{\alpha}, t)$ is the trajectory of the fluid particle which at time $t = 0$ is at the point $\boldsymbol{\alpha}$. For fixed $\boldsymbol{\alpha}$ this trajectory may be found by solving the ordinary differential equation,

$$\frac{d\mathbf{x}}{dt}(\boldsymbol{\alpha}, t) = \mathbf{u}(\mathbf{x}(\boldsymbol{\alpha}, t), t)\,, \qquad (11.3.1a)$$

$$\mathbf{x}(\boldsymbol{\alpha}, 0) = \boldsymbol{\alpha}\,. \qquad (11.3.1b)$$

In this section we will assume that $\Omega = I\!R^n$ and substitute the free-space boundary condition (11.2.2) for the no-flow boundary condition (11.2.1d). In this case $\mathbf{u}$ is a solution of the incompressible Euler equations (11.2.1a–c) with boundary condition (11.2.2). Therefore, by (11.2.7) we have

$$\mathbf{u}(\mathbf{x}, t) = \int_{I\!R^n} \mathbf{K}(\mathbf{x} - \mathbf{x}')\, \boldsymbol{\omega}(\mathbf{x}', t)\, d\mathbf{x}' = \int_{I\!R^n} \mathbf{K}(\mathbf{x} - \mathbf{x}(\boldsymbol{\alpha}', t))\, \boldsymbol{\omega}(\mathbf{x}(\boldsymbol{\alpha}', t), t)\, d\boldsymbol{\alpha}'\,.$$

The last equality follows from the fact that $\nabla \cdot \mathbf{u} = 0$ and hence, that the Jacobian of the transformation $\boldsymbol{\alpha}' \rightarrow \mathbf{x}(\boldsymbol{\alpha}', t)$ equals 1 (e.g., see Chorin and Marsden [1990], p. 10).

Thus, (11.3.1a,b) can be rewritten as

$$\frac{d\mathbf{x}}{dt}(\boldsymbol{\alpha},t) = \int_{I\!R^n} \mathbf{K}(\mathbf{x}(\boldsymbol{\alpha},t) - \mathbf{x}(\boldsymbol{\alpha}',t))\, \boldsymbol{\omega}(\mathbf{x}(\boldsymbol{\alpha}',t),t)\, d\boldsymbol{\alpha}', \tag{11.3.2a}$$

$$\mathbf{x}(\boldsymbol{\alpha},0) = \boldsymbol{\alpha}. \tag{11.3.2b}$$

Thus, if the fluid flow map $\boldsymbol{\alpha} \rightarrow \mathbf{x}(\boldsymbol{\alpha},t)$ and the vorticity field $\boldsymbol{\omega}(\mathbf{x},t)$ are known, then the velocity at later times may be written as an integral over the domain at the initial time $t = 0$. In two dimensions the vorticity is constant on particle paths $\boldsymbol{\alpha} \rightarrow \mathbf{x}(\boldsymbol{\alpha},t)$ and this leads to an especially simple representation of the velocity field at later times.

11.3.1 Two Dimensions

As a consequence of the scalar character of the vorticity in two dimensions the two dimensional version of the vortex method is simpler than its three dimensional counterpart. Therefore we begin by describing the vortex method in $I\!R^2$. In two dimensions the fluid velocity $\mathbf{u}$ is perpendicular to the vorticity

$$(\boldsymbol{\omega} \cdot \nabla)\mathbf{u} = 0.$$

Hence, the right hand side of (11.2.13a) is zero,

$$\frac{D\omega}{Dt} = 0.$$

This equation states that the vorticity is transported passively by the flow. In other words, that the vorticity is constant along particle trajectories,

$$\frac{d}{dt}\omega(\mathbf{x}(\boldsymbol{\alpha},t),t) = 0,$$

or equivalently,

$$\omega(\mathbf{x}(\boldsymbol{\alpha},t),t) = \omega^0(\boldsymbol{\alpha}). \tag{11.3.3}$$

This is the most significant difference between the fluid flow equations in two and three dimensions. In general, when $n = 3$ we have $(\boldsymbol{\omega} \cdot \nabla)\mathbf{u} \neq 0$. This provides a mechanism for the vorticity to change as it moves with the flow. In particular, the vorticity may now stretch and fold. It is generally believed that this is the mechanism by which turbulence is created and driven (e.g., see Tennekes and Lumley [1972]). This explains why the phrase "two dimensional turbulence" is sometimes considered an oxymoron.

Using (11.3.3) we can rewrite (11.3.2a,b) as

$$\frac{d\mathbf{x}}{dt}(\boldsymbol{\alpha},t) = \int_{I\!R^n} \mathbf{K}(\mathbf{x}(\boldsymbol{\alpha},t) - \mathbf{x}(\boldsymbol{\alpha}',0))\, \omega^0(\boldsymbol{\alpha}')\, d\boldsymbol{\alpha}', \tag{11.3.4a}$$

$$\mathbf{x}(\boldsymbol{\alpha},0) = \boldsymbol{\alpha}. \tag{11.3.4b}$$

Thus, given the fluid flow map $\boldsymbol{\alpha} \longrightarrow \mathbf{x}(\boldsymbol{\alpha}, t)$, we see that the velocity can be expressed as an integral over the vorticity field at time $t = 0$. The equivalence of the Lagrangian equations (11.3.4a,b) and the Euler equations (11.2.1a–c) with boundary condition (11.2.2) is discussed in McGrath [1968].

In the vortex method the solution of (11.3.4a,b) is approximated by discretizing the integral on the right hand side of (11.3.4a) and solving the resulting equations for a finite number of particles. Consider a rectangular grid with mesh width h that covers the support of $\omega^0 = \nabla \times \mathbf{u}^0$.[2] Let $\boldsymbol{\alpha}_i$, $i = 1, \ldots, N$, denote the centers of the grid cells and let $\omega_i = \omega(\boldsymbol{\alpha}_i)$. If we denote by $\tilde{\mathbf{x}}_i(t)$ the trajectory of the particle that starts at $\boldsymbol{\alpha}_i$ and evolves according to this discretized system, then an obvious approximation to (11.3.4a,b) consists of solving the following system of N ordinary differential equations,

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \sum_{j \neq i} \mathbf{K}(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \omega_j\, h^2\,, \tag{11.3.5a}$$

$$\tilde{\mathbf{x}}_i(0) = \boldsymbol{\alpha}_i\,. \tag{11.3.5b}$$

The numerical solution of this system is known as the *point vortex method.* This method was originally used by Rosenhead [1931] for computing the roll-up of a vortex sheet. It is apparent from (11.2.8) that whenever two particle trajectories approach each other, the velocity that each induces on the other goes to infinity. For this reason it was generally believed that the point vortex method was unstable and would not converge to solutions of the incompressible Euler equations. However, Goodman, Hou, and Lowengrub [1990a] have recently proved that the 2-d point vortex method does in fact converge to solutions of (11.3.4a,b). We will briefly discuss this and related work in Section 11.3.5.

Chorin [1973] introduced the idea of replacing (11.3.5a,b) by

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \sum_{j \neq i} \mathbf{K}_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \omega_j\, h^2, \tag{11.3.6a}$$

$$\tilde{\mathbf{x}}_i(0) = \boldsymbol{\alpha}_i\,. \tag{11.3.6b}$$

where the new kernel $\mathbf{K}_\delta$ is close to $\mathbf{K}$ except at the origin where $\mathbf{K}_\delta$ is bounded. From a theoretical point of view Hald has shown that it is convenient to obtain $\mathbf{K}_\delta$ by convoluting $\mathbf{K}$ with a *smoothing function* f_δ. So $\mathbf{K}_\delta = \mathbf{K} * f_\delta$ where $f_\delta : \mathbb{R}^2 \longrightarrow \mathbb{R}$ is a scalar function on $\mathbb{R}^2$ which is obtained from a fixed scalar function $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ of integral one via the relation

$$f_\delta(\mathbf{x}) = \frac{1}{\delta^2} f(\mathbf{x}/\delta)\,. \tag{11.3.7}$$

[2] For simplicity we always assume that ω^0 has compact support. With suitable modifications one can also handle initial vorticity distributions ω^0 that are periodic or that have unbounded support but decay sufficiently fast at infinity.

If we compare (11.3.6a) with (11.3.1a), we find that solving the system (11.3.6a,b) amounts to replacing the exact velocity $\mathbf{u}(\mathbf{x}, t)$ in (11.3.1a) by the approximate velocity

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_i \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i(t))\, \omega_i\, h^2 \,, \tag{11.3.8}$$

and computing the N approximate particle trajectories $\tilde{\mathbf{x}}_i$ that start at the points $\boldsymbol{\alpha}_i$. The discretization presented here is essentially a trapezoid rule approximation to the right hand side of (11.3.4a). Nicolaides [1986] and Chiu and Nicolaides [1988] have studied other discretizations of this integral.

Since by definition $\mathbf{K}_\delta = \mathbf{K} * f_\delta$ and since $\nabla \times (\mathbf{K} * g) = g$ for any scalar function g which decays sufficiently fast as $|\mathbf{x}| \to \infty$, it follows from taking the curl of equation (11.3.8) that our approximation to the vorticity field is

$$\tilde{\omega}(\mathbf{x}, t) = \sum_i f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i(t))\, \omega_i\, h^2 \,. \tag{11.3.9}$$

The ith term on the right hand side of (11.3.9) is referred to as the ith *vortex* or ith *vortex blob*, $\tilde{\mathbf{x}}_i(t)$ is its position at time t and ω_i is its *strength* or *weight*.

The smoothness of $\mathbf{K}_\delta$ depends on f_δ. We can choose f_δ so that the kernel $\mathbf{K}_\delta$ will be bounded or even continuous and differentiable at $\mathbf{x} = 0$. Chorin [1973] originally used

11.3.34

$$f_\delta(\mathbf{x}) = \begin{cases} (2\pi|\mathbf{x}|\delta)^{-1} & |\mathbf{x}| < \delta, \\ 0 & |\mathbf{x}| \geq \delta. \end{cases} \tag{11.3.10}$$

This yields a velocity of the form (11.3.8) with

$$\mathbf{K}_\delta(\mathbf{x}) = (-y,\, x)^T \begin{cases} (2\pi|\mathbf{x}|\delta)^{-1} & |\mathbf{x}| < \delta, \\ (2\pi|\mathbf{x}|^2)^{-1} & |\mathbf{x}| \geq \delta. \end{cases} \tag{11.3.11}$$

This has probably been the most widely used velocity kernel in two dimensional "engineering" implementations of the vortex method. By comparing (11.3.11) with (11.2.8) we see that $\mathbf{K}_\delta(\mathbf{x}) = \mathbf{K}(\mathbf{x})$ for $\mathbf{x} \geq \delta$ but that $\mathbf{K}_\delta$ remains bounded as $\mathbf{x} \to 0$. Thus, the effect of f_δ is to "cutoff" the infinite velocity at the origin. For this reason the function f_δ is also referred to as the *cutoff function* and δ is called the *cutoff radius*. For f_δ given by (11.3.10) it is apparent from (11.3.9) that the vorticity due to a vortex blob at $\mathbf{x}_i$ is now concentrated in a circular region or "core" about $\mathbf{x}_i$. Hence f_δ is sometimes also called a *core function*.

Usually – but not always – f is chosen to be radially symmetric: $f(\mathbf{x}) = f(r)$ where $r = |\mathbf{x}|$. It is sometimes the case, especially for higher order cutoffs, that f does not have compact support. In these cases the name cutoff radius for δ no longer quite makes sense. For this reason some authors prefer to refer to δ as the *cutoff parameter*. Assuming that the exact solution $\mathbf{u}(\mathbf{x},t)$ of (11.2.1a–c) with boundary condition (11.2.2) is sufficiently smooth, then the properties of f will determine the accuracy of the vortex method approximation to $\mathbf{u}$. We discuss cutoff functions further in Section 11.3.3 below.

Suppose that for some $R_0 > 0$ we have $\mathbf{K}_\delta(\mathbf{x}) = \mathbf{K}(\mathbf{x})$ whenever $|\mathbf{x}| \geq R_0$. (In Section 11.3.3 we will show that this is true for radially symmetric f with compact support.) Let C be any circle of radius $R > R_0$ centered at $\tilde{\mathbf{x}}_i(t)$. Then the circulation Γ_i about the ith vortex is

$$\begin{aligned}\Gamma_i &= \oint_C \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i(t))\,\omega_i\,h^2 \cdot \mathbf{ds}\\ &= \frac{\omega_i h^2}{2\pi} \oint_C \frac{(-y,x)}{|\mathbf{x}|^2} \cdot \mathbf{ds}\\ &= \frac{\omega_i h^2}{2\pi} \int_0^{2\pi} (-\sin\theta, \cos\theta)\cdot(-\sin\theta,\cos\theta)\,d\theta\\ &= \omega_i h^2\,.\end{aligned} \tag{11.3.12}$$

This formula is useful for determining the vortex strengths when vortices can enter the computation by some mechanism other than as part of the initial discretization of the vorticity field. This occurs most often within the context of satisfying the no-slip boundary condition (11.2.15) on $\partial\Omega$. We will discuss such techniques in Section 11.6. It is also possible to design a vortex method in which vortices are created at a flame front. Such algorithms have been studied by Ashurst and McMurtry [1989], Ghoniem, Chorin, and Oppenheim [1982] and Sethian [1984].

Equations (11.3.6a,b) are a system of N ordinary differential equations that describe the approximate particle trajectories $\tilde{\mathbf{x}}_i(t)$, each of which is continuous in the variable t. One hopes that for appropriately chosen f_δ these approximate particle trajectories remain close to the exact particle trajectories $\mathbf{x}(\alpha_i, t)$. This turns out to be true and can be rigorously proven as we shall see in Section 11.3.4.

In an actual computation the trajectories $\tilde{\mathbf{x}}_i(t)$ are further approximated by discretizing (11.3.6a,b) in time. One can do this by employing almost any numerical ODE solver. For example, let $\tilde{\mathbf{x}}_i^k$ denote the time discretized approximation to $\tilde{\mathbf{x}}_i(t)$ at time $t_k = k\Delta t$ and let

$$\tilde{\mathbf{u}}^k(\mathbf{x}) = \sum_i \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i^k)\,\omega_i\,h^2\,, \tag{11.3.13}$$

denote the approximate velocity field. If we use the first order Euler's method to approximate (11.3.6a,b) in time, then the new particle positions at time $t_{k+1} = (k+1)\Delta t$ are given by,

$$\tilde{\mathbf{x}}_i^{k+1} = \tilde{\mathbf{x}}_i^k + \Delta t\,\tilde{\mathbf{u}}^k(\tilde{\mathbf{x}}_i^k)\,. \tag{11.3.14}$$

Thus, since the vorticity is constant on particle paths, the approximate velocity field at time t_{k+1} is now,

$$\tilde{\mathbf{u}}^{k+1}(\mathbf{x}) = \sum_i \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i^{k+1})\, \omega_i\, h^2 \, .$$

Higher order time discretizations have been studied theoretically by Anderson and Greengard [1985] and Hald [1987]. Examples of computations which involve a high order time discretization may be found in Baden and Puckett [1990], Cheer [1989], Nordmark [1991], Sethian [1984], and Sethian and Ghoniem [1988].

Note that since the vortex strengths ω_i have not changed on the approximate particle paths $\tilde{\mathbf{x}}_i^k \to \tilde{\mathbf{x}}_i^{k+1}$ we are approximately satisfying (11.2.13a) (or equivalently (11.3.3)). Furthermore, since $\tilde{\mathbf{u}}^{k+1} = \mathbf{K} * \tilde{\omega}^{k+1}$ explicitly satisfies (11.2.13b,c), the new velocity field $\tilde{\mathbf{u}}^{k+1}$ is a consistent approximation to the exact solution of (11.2.13a–c) at time t_{k+1} with initial data $\tilde{\omega}^k$ at time $t_k = k\Delta t$. Finally, for smoothing functions f_δ which have compact support or which decay sufficiently fast at infinity, the approximate velocity field $\tilde{\mathbf{u}}^k$ given by (11.3.13) will automatically satisfy the free space boundary condition (11.2.2).

A Remark Concerning Notation We denote the exact trajectory of the particle which at time $t = 0$ is at the point $\boldsymbol{\alpha}$ by $\mathbf{x}(\boldsymbol{\alpha}, t)$. In other words, $\mathbf{x}(\boldsymbol{\alpha}, t)$ is a solution of the exact Lagrangian equations (11.3.4a,b). If we are considering a finite number of exact particle trajectories, say starting at the points $\boldsymbol{\alpha}_i$, $i = 1, \ldots, N$, then we simply write $\mathbf{x}_i(t)$. So the $\mathbf{x}_i(t)$ denote solutions of the finite set of ODEs

$$\frac{d\mathbf{x}_i}{dt}(t) = \int_{\mathbb{R}^n} \mathbf{K}(\mathbf{x}_i(t) - \mathbf{x}(\boldsymbol{\alpha}, t))\, \omega^0(\boldsymbol{\alpha})\, d\boldsymbol{\alpha}\, ,$$
$$\mathbf{x}_i(0) = \boldsymbol{\alpha}_i\, ,$$

for $i = 1, \ldots, N$, where the flow map $\mathbf{x}(\boldsymbol{\alpha}, t)$ is assumed to be known.

We denote a (time continuous) approximation of the ith particle trajectory $\mathbf{x}_i(t)$ by $\tilde{\mathbf{x}}_i(t)$. The paths $\tilde{\mathbf{x}}_i(t)$ are determined by solving the ODEs (11.3.6a,b). We denote a time discretized approximation to the $\tilde{\mathbf{x}}_i(t)$ at time t_k by $\tilde{\mathbf{x}}_i^k$. For example, if we use the first order Euler's method, then the $\tilde{\mathbf{x}}_i^k$ are advanced in time according to (11.3.14) with $\tilde{\mathbf{u}}^k$ defined by (11.3.13). We will employ a similar notation for the three dimensional versions of these quantities.

The purpose of distinguishing between the time continuous approximation $\tilde{\mathbf{x}}_i(t)$ to $\mathbf{x}_i(t)$ and the time discretized version $\tilde{\mathbf{x}}_i^k$ is as follows. Historically the first convergence proofs for the vortex method were for the time continuous case. In other words, it was shown that the $\tilde{\mathbf{x}}_i(t)$ converge to the $\mathbf{x}_i(t)$ and that the corresponding velocity field $\tilde{\mathbf{u}}(\mathbf{x}, t)$ converges to the exact velocity field $\mathbf{u}(\mathbf{x}, t)$ as the discretization of the right hand side of (11.3.2a) is refined. Subsequently, it was also proved that the $\tilde{\mathbf{x}}_i^k$ converge to $\mathbf{x}_i(t_k)$ and that $\tilde{\mathbf{u}}^k(\mathbf{x})$ converges to $\mathbf{u}(\mathbf{x}, t_k)$ in the appropriate limit. However many workers continue to state and prove theorems for only time continuous versions of the vortex method. In particular, there has not yet been published a convergence proof for the time discretized version of the random vortex method in three dimensions. By emphasizing the distinction between the time discretized and time continuous versions of the vortex method we will be able to give precise statements of some of these theorems. It is also our hope that by making the distinction clear here this will assist readers that go on to read other papers in the field.

We have neglected one essential issue in our discussion of the vortex method in two dimensions. This is the problem of satisfying the no-flow boundary condition (11.2.1d) in a computation. We will discuss this in Section 11.4. We now describe the vortex method for approximating solutions of (11.2.1a–c) with boundary condition (11.2.2) in three dimensions.

11.3.2 Three Dimensions

Recall that in $I\!R^3$ the vorticity is a vector which may point in any direction. This implies that the right hand side of (11.2.13a) is in general nonzero. This term is usually associated with the "stretching" of vorticity. One can think of (11.2.13a) as an evolution equation for the vorticity in which the vorticity is stretched by the flow as it is transported along particle trajectories. To see this note that by (11.2.13a) the evolution of vorticity that is carried with the point $\mathbf{x}(\boldsymbol{\alpha}, t)$ is given by

$$\frac{d\boldsymbol{\omega}}{dt}(\mathbf{x}(\boldsymbol{\alpha}, t), t) = (\boldsymbol{\omega}(\mathbf{x}(\boldsymbol{\alpha}, t), t) \cdot \nabla)\, \mathbf{u}(\mathbf{x}(\boldsymbol{\alpha}, t), t)\,. \tag{11.3.15}$$

Hence the stretching term $(\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u}$ governs the rate of change of the vorticity as it is convected with the flow.

A three dimensional vortex method also involves the discretization of (11.3.2a,b). However, now the vorticity $\boldsymbol{\omega}(\mathbf{x}, t)$ is a vector quantity and it is no longer constant along particle paths. As in two dimensions we discretize the right hand side of (11.3.2a) by approximating the velocity field by a sum of the form

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_i \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i(t))\, \tilde{\boldsymbol{\omega}}_i(t)\, h^3\,. \tag{11.3.16}$$

Here $\tilde{\boldsymbol{\omega}}_i(t)$ is an approximation of $\boldsymbol{\omega}(\tilde{\mathbf{x}}_i(t), t)$, $\mathbf{K}_\delta = \mathbf{K} * f_\delta$ with $\mathbf{K}$ given by (11.2.9), and f_δ is a smoothing function of the form

$$f_\delta(\mathbf{x}) = \frac{1}{\delta^3} f(\mathbf{x}/\delta)\,, \tag{11.3.17}$$

for some $f : I\!R^3 \to R$ of integral one. However, unlike the two dimensional case, we must now find a way to update the vortex strengths $\omega_i(t)$. A variety of techniques have been proposed for doing this. Essentially three dimensional vortex methods are distinguished from one another by the manner in which they discretize equation (11.3.15).

In this article we will be primarily concerned with a method due to Anderson which was first described in Anderson and Greengard [1985]. In this method one uses (11.3.16) to find an expression for $\nabla\mathbf{u} \equiv (\nabla u, \nabla v, \nabla w)$ of the form

$$\nabla\tilde{\mathbf{u}}(\mathbf{x},t) = \nabla \sum_j \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3\,. \tag{11.3.18}$$

Since $\mathbf{K}_\delta$ is a known function of $\mathbf{x}$, the right hand side of (11.3.18) can be explicitly computed. This suggests the following system of $2N$ ordinary differential equations for the vortex positions and strengths

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \sum_j \mathbf{K}_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3\,, \tag{11.3.19a}$$

$$\tilde{\mathbf{x}}_i(0) = \boldsymbol{\alpha}_i\,, \tag{11.3.19b}$$

$$\frac{d\tilde{\boldsymbol{\omega}}_i}{dt}(t) = (\tilde{\boldsymbol{\omega}}_i(t)\cdot\nabla) \sum_j \mathbf{K}_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3\,, \tag{11.3.20a}$$

$$\tilde{\boldsymbol{\omega}}_i(0) = \boldsymbol{\omega}^0(\boldsymbol{\alpha}_i)\,. \tag{11.3.20b}$$

Note that equations (11.3.20a,b) describe the evolution of the vortex strengths $\tilde{\omega}_i(t)$ in time rather than the evolution of the vorticity field itself. Nevertheless, it is natural to expect that the solution of these equations converges to the exact solution of the 3D incompressible Euler equations in the limit as $h,\, \delta \to 0$. This has been proved by Beale [1986] and Cottet [1988b] under certain reasonable assumptions on the smoothness of the underlying flow field and the choice of the smoothing function f_δ. In addition, Long [1990] has proved that a version of the random vortex method based on this method converges to solutions of the 3-D Navier-Stokes equations. We will discuss these results further in Sections 11.3.5 and 11.5.3.

A Method of Beale and Majda An alternative discretization of (11.3.15) was proposed by Beale and Majda [1982a]. It is based on the fact that equation (11.2.13a) implies

$$\boldsymbol{\omega}(\mathbf{x}(\boldsymbol{\alpha},t),t) = (\nabla_\alpha \mathbf{x}(\boldsymbol{\alpha},t))\, \boldsymbol{\omega}^0(\boldsymbol{\alpha})\,,$$

(e.g., see Chorin and Marsden [1990]). This suggests the following method for updating the vortex strengths

$$\tilde{\boldsymbol{\omega}}_i(t) = (\nabla^h_\alpha \tilde{\mathbf{x}}(\boldsymbol{\alpha}_i, t)) \cdot \boldsymbol{\omega}^0(\boldsymbol{\alpha}_i)\,, \qquad i = 1, \ldots, N\,, \tag{11.3.21}$$

where ∇^h_α denotes some (high order accurate) finite difference approximation to the gradient of the fluid flow map which is defined on the grid determined by the vortex positions at time $t = 0$. It has been proved by Beale and Majda [1982a and 1982b] and Anderson and Greengard [1985] that the solution of the coupled system of equations (11.3.19a,b) and (11.3.21) also converges to the solution of the 3-d incompressible Euler equations in the appropriate limit.

The Vortex Filament Method We mention one other approach, originally proposed by Chorin [1980 and 1982], which is sometimes called the *vortex filament method.* In this method the vorticity field is represented as a collection of vectors. Each vector is represented by two points, considered to be at the head and tail of the vector, and the magnitude of the vorticity is proportional to the distance between these two points. The evolution of this vorticity field is approximated by allowing these points to flow with the velocity field induced by this approximate vorticity field via (11.2.7). The stretching term is automatically accounted for by the movement of these two points in relation to one another. (In an actual computation one usually divides a vector into two smaller vectors when the distance between the two points becomes large.) Greengard [1986] observed that the vortex filament method is essentially equivalent to the method of Beale and Majda and used this observation to prove that the method converges to solutions of the incompressible Euler equations under the appropriate conditions. The interested reader is also referred to the work of Anderson and Greengard [1985 and 1989] for further information concerning the vortex filament method.

These are just a few examples of a wide variety methods that have been suggested for updating the vortex strengths in three dimensions. We refer the reader to the literature for a more detailed discussion of this topic: Ashurst and Meiburg [1988], Anderson and Greengard [1985], Beale and Majda [1982a and 1982b], Fishelov [1990a], Greengard [1986], Knio and Ghoniem [1990] and Leonard [1980 and 1985]. We also remark that Beale, Eydeland, and Turkington [1991] have conducted numerical studies to compare several of the algorithms described above.

In an actual computation one must discretize the solution of (11.3.19a,b) and (11.3.20a,b) (or (11.3.21)) in time. As in two dimensions let $\tilde{\mathbf{x}}^k_i$ denote the time discretized approximation to $\tilde{\mathbf{x}}_i(t_k)$ and $\tilde{\mathbf{u}}^k$ denote the approximation to $\tilde{\mathbf{u}}(\mathbf{x}, t_k)$. The time discretized velocity field at time t_k may therefore be written as

$$\tilde{\mathbf{u}}^k(\mathbf{x}) = \sum_i \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i^k)\, \tilde{\omega}_i^k\, h^3\,.$$

If we wish to approximate (11.3.19a,b) to first order in time, then the particle positions are advanced according to (11.3.14), just as in the two dimensional case. Similarly, a first order time discretization of (11.3.20a,b) is given by

$$\boldsymbol{\omega}_i^{k+1} = \boldsymbol{\omega}_i^k + \Delta t\, (\boldsymbol{\omega}_i^k \cdot \nabla)\, \tilde{\mathbf{u}}^k(\mathbf{x}_i^k)\,. \qquad (11.3.22)$$

A Remark Concerning the Approximate Vorticity Field in Three Dimensions In analogy with the two dimensional case it is natural to let the approximate vorticity field be defined by

$$\mathbf{vort}(\mathbf{x}, t) = \sum_i f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i(t))\, \tilde{\omega}_i(t)\, h^3\,. \qquad (11.3.23)$$

However, in general this function fails to be divergence free. (In two dimensions the fact that the vorticity is a vector pointing in a direction perpendicular to the x, y-plane guarantees that (11.3.9) is divergence free.) It is apparent then that **vort** $\neq \nabla \times \tilde{\mathbf{u}}$. Instead one can define

$$\tilde{\omega}(\mathbf{x}, t) = \nabla \times \tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_i \zeta_\delta(\mathbf{x} - \tilde{\mathbf{x}}_i^k)\, \tilde{\omega}_i^k\, h^3\,, \qquad (11.3.24)$$

where $\zeta_\delta \equiv \nabla \times \mathbf{K}_\delta$. (Hald [1991] has taken the trouble to actually write out ζ_δ.)

As Beale [1986] has pointed out, $\tilde{\omega}$ is simply the L^2 projection of **vort** onto the space of divergence free vector fields. In particular, the velocity field induced by (11.3.23) is identical to that induced by (11.3.24). Since in an actual computation one does not need to explicitly evaluate the vorticity field in order to evolve the flow field, the question of which of the above expressions represents the computed vorticity field is largely academic. However, if one is interested in computing values of the vorticity field in addition to the velocity (say, for example, in order to plot the vorticity field) or if one is interested in proving that the computed vorticity field converges to the exact one, then it may very well be appropriate to use (11.3.24) rather than (11.3.23). We refer the interested reader to the comments of Anderson and Greengard [1989, p. 1129] and Beale [1986, p. 405] for a more detailed discussion of this issue.

11.3.3 Initial Conditions

We now briefly discuss how to choose the initial vortex positions and strengths. Suppose we are given an initial vorticity field $\boldsymbol{\omega}^0(\mathbf{x})$ with support contained in a bounded set $\Omega_0 \subseteq I\!\!R^n$. We wish to approximate $\boldsymbol{\omega}^0$ by a sum of the form (11.3.9), (11.3.23), or (11.3.24). We begin by creating a square grid of side h that covers Ω_0. We let the initial vortex positions $\tilde{\mathbf{x}}_i^0 = \tilde{\mathbf{x}}_i(0)$ be the centers of the grid cells and the initial particle strengths be

$$\omega_i^0 = \omega(\tilde{\mathbf{x}}_i^0)\,. \tag{11.3.25}$$

This is the type of initial condition that is assumed in most proofs that the vortex method (*respectively* random vortex method) converges to solutions of Euler's (*respectively* Navier-Stokes) equations. (However in many computations - especially viscous ones – vorticity may introduced into the flow via other mechanisms. We will discuss this further in Section 11.5.2 below.) In this context the grid spacing h plays a central role in the accuracy of the method. It, together with the cutoff radius δ, determines the accuracy of the initial discretization and as well as the accuracy of the subsequent approximation $\tilde{\mathbf{u}}^k$ to the flow field at later times.

It is also possible to choose the initial particle strengths so that

$$\omega_i^0\, h^n = \int_{B_i} \omega(\mathbf{x})\, d\mathbf{x}\,, \tag{11.3.26}$$

where B_i denotes the ith grid cell. For example, Hald [1979] used this initial condition in his convergence proof. In fact, (11.3.26) may be preferable to (11.3.25) when the initial data is not very smooth. We also note that Anderson and Greengard [1985], Nicolaides [1986], and Chiu and Nicolaides [1988] have studied how to choose the vortex positions and strengths on irregular grids in a manner that will preserve the overall accuracy of the method.

11.3.4 Cutoff Functions

We now turn to a discussion the cutoff function f_δ. As we have already noted the properties of f, together with the smoothness of the underlying flow field, will determine the accuracy of the vortex method. In this section we discuss some of the properties necessary for an accurate smoothing function and describe some that have been proposed. In the following section we state some of the convergence results that have been obtained by researchers in the field. These results will illustrate how the choice of cutoff function influences the rate of convergence. Sections 11.3.4 and 11.3.5 are loosely based on Hald's [1991] review article.

In what follows C denotes a generic constant, D denotes differentiation, $\beta = (\beta_1, \ldots, \beta_n)$ is a multi-index the components of which are non-negative integers, and $|\beta| = \beta_1 + \ldots + \beta_n$. So for example, when $n = 2$ we have,

$$\mathbf{x}^\beta = x^{\beta_1} y^{\beta_2}\,, \qquad \text{and} \qquad D^\beta f = \frac{\partial^{|\beta|} f}{\partial_x^{\beta_1} \partial_y^{\beta_2}}\,,$$

for any function $f : I\!R^2 \rightarrow I\!R$.

We remind the reader that given $f : I\!R^n \rightarrow I\!R$, the function f_δ is defined by (11.3.7) in two dimensions or (11.3.17) in three dimensions. There have been a variety of assumptions placed on the smoothing function f_δ in the literature. For example, in the first convergence proof of any kind for the vortex method Hald and Del Prete [1978] assumed that f_δ was chosen so that $\mathbf{K}_\delta(\mathbf{x}) = \mathbf{K} * f_\delta(\mathbf{x})$ is continuous for $\mathbf{x} \neq 0$ and that for all $|\beta| \leq 2$ the derivatives of $\mathbf{K}_\delta$ satisfy

$$\left|D^\beta \mathbf{K}_\delta(\mathbf{x})\right| \leq \begin{cases} C\,\delta^{-1}|\mathbf{x}|^{-|\beta|} & 0 < |\mathbf{x}| \leq \delta, \\ C\,|\mathbf{x}|^{-|\beta|-1} & \delta < |\mathbf{x}|. \end{cases}$$

(For example, Chorin's cutoff (11.3.11) satisfies these assumptions.) Given the above assumptions on $\mathbf{K}_\delta$ Hald and Del Prete were able to prove first order convergence of the two dimensional vortex method for short times.

By strengthening these assumptions Hald [1979] was subsequently able to establish second order convergence of the two dimensional vortex method for arbitrarily long times. In this work Hald made the following assumptions on the smoothing function: f_δ is radially symmetric $f_\delta(\mathbf{x}) = f_\delta(r)$ where $r = |\mathbf{x}|$, $f_\delta(r)$ vanishes for $r \geq \delta$, $f_\delta(r)$ is twice continuously differentiable with respect to r, f_δ'' is continuously differentiable except possibly at $r = \delta$, $f_\delta'(0) = 0$,

$$\int_0^\delta r f_\delta(r)\, dr = \frac{1}{2\pi}\,,$$
$$\int_0^\delta r^3 f_\delta(r)\, dr = 0\,,$$

and for $p \leq 3$ the derivatives of f_δ can be estimated by

$$\left|\frac{d^p}{dr^p} f_\delta(r)\right| \leq \begin{cases} C\delta^{-2-p} & r \leq \delta, \\ 0 & r > \delta. \end{cases}$$

In order to obtain higher order convergence still more more stringent requirements on f_δ are necessary. The first general theory of high order convergence for the vortex method is due to Beale and Majda [1982b]. In this paper the authors defined a class of smoothing functions called $FeS^{-L,p}$ and proved that for $f \in FeS^{-L,p}$ both two and three dimensional versions of the vortex method converge to solutions of the Euler equations at an arbitrarily high rate. The order of convergence depends only on the smoothness of the flow and the appropriate choice of L and p. (The parameters L and p here correspond to the parameters L and m defined below.) Cottet [1982] and Raviart [1985] have considered similar classes in their work. However, rather than compare various classes of smoothing functions we pick one and briefly discuss the nature of the various requirements. The following class, called $M^{L,m}$, is due to Anderson and Greengard [1985].[3]

Definition: Let $f : I\!R^n \to I\!R$ be a scalar function defined on $I\!R^n$ for $n = 2$ or 3. For $L \geq 3$ we say that the function $f \in M^{L,m}$ if f is L times continuously differentiable and satisfies the following properties,

$$\int_{I\!R^n} f(\mathbf{x})\, d\mathbf{x} = 1\,, \tag{11.3.27}$$

$$\int_{I\!R^n} \mathbf{x}^\beta f(\mathbf{x})\, d\mathbf{x} = 0\,, \qquad \text{for all } 1 \leq |\beta| \leq m-1\,, \tag{11.3.28}$$

$$\int_{I\!R^n} |\mathbf{x}|^m\, |f(\mathbf{x})|\, d\mathbf{x} < \infty\,, \tag{11.3.29}$$

$$|\mathbf{x}|^{n+|\beta|}\, |D^\beta f(\mathbf{x})| \leq C\,, \qquad \text{for all } |\beta| \leq L\,, \tag{11.3.30}$$

$$|\mathbf{x}|^{m+n+2}\, |f(\mathbf{x})| \leq C\,. \tag{11.3.31}$$

Given $f \in M^{L,m}$ we define

$$f_\delta(\mathbf{x}) = \frac{1}{\delta^n} f(\mathbf{x}/\delta)\,. \tag{11.3.32}$$

Functions f_δ with $f \in M^{L,m}$ are called mth order cutoff functions. The justification for this terminology will become apparent in the next section.

From (11.3.27) and (11.3.32) it follows that the family of functions $\{f_\delta : \delta > 0\}$ is an approximation to the identity (e.g., see Folland [1976, p. 17]). In other words, for all $g \in \mathcal{L}^p$, with $1 \leq p < \infty$,

$$f_\delta * g \to g \qquad \text{as} \qquad \delta \to 0\,,$$

in the $\mathcal{L}^p$ norm. Thus we may regard f_δ as an approximation to the Dirac delta function.

Condition (11.3.28) is what has become known as the *moment condition.* It states that all moments of the function f up to order $m-1$ vanish. It is natural to choose f to be radially symmetric. In this case all of the odd moments of f vanish by symmetry and it follows that m is even. In particular, if f is radially symmetric and no moment conditions are imposed, then $m = 2$. Conditions (11.3.27–11.3.29) ensure that replacing the singular kernel $\mathbf{K}$ by the smoothed kernel $\mathbf{K}_\delta = \mathbf{K} * f_\delta$ in (11.2.7) causes a change in the velocity which is $O(\delta^m)$ (e.g., see Anderson and Greengard [1985, p. 422]).

[3] The M stands for "mollifier."

Conditions (11.3.29–11.3.31) govern the decay of f at infinity. These are essentially technical requirements that are used in the convergence theory to ensure that the remainder in certain integrals is easy to estimate. A variety of similar conditions have been considered in the literature. For instance, Beale and Majda [1982b] originally considered bounds on the Fourier transform of f rather than (11.3.29–11.3.31). In general, one needs f to be "smooth and rapidly decreasing." For example, in Beale and Majda [1985] the authors replace (11.3.29–11.3.31) with

$$|D^{\beta} f(\mathbf{x})| \leq C\,(1+|\mathbf{x}|^2)^{-j}\,, \tag{11.3.33}$$

for all multi-indices β and every integer j. Conditions (11.3.27–11.3.28) together with (11.3.33) are perhaps the simplest requirements one can place on f. However (11.3.33) is a bit more restrictive than is needed in the theory. Finally we note that if one only considers functions f with compact support, then (11.3.29–11.3.31) can be omitted altogether. Choosing f to have compact support may also be desirable for other reasons which we discuss at the end of this section.

For radially symmetric functions $f(\mathbf{x}) = f(r)$, one can show that

$$\mathbf{K}_\delta(\mathbf{x}) = \mathbf{K}(\mathbf{x}) \int_{|\mathbf{y}|\leq r} f_\delta(\mathbf{y})\,d\mathbf{y} = \mathbf{K}(\mathbf{x})\;F(r)\,, \tag{11.3.34}$$

(e.g., see Hald [1979], p. 735). The function $F(r) = 2\pi \int_0^r s f_\delta(s)\,ds$ is called the the *shape factor* or sometimes (somewhat confusingly) the *cutoff function*. It is apparent from (11.3.27) and (11.3.34) that if f has compact support, say $f(r)=0$ for $r > a$, then $\mathbf{K}_\delta$ agrees with $\mathbf{K}$ for $r > a\delta$. We also note that if f_δ is radially symmetric, then $\mathbf{K}_\delta$ can be constructed from f_δ and $\mathbf{K}$ by using (11.3.34).

A wide variety of explicit cutoff functions of various orders of accuracy have appeared in the literature. We list a few of these here. Beale and Majda [1985] studied cutoffs based on the Gaussian. For example, in two dimensions

$$f_2(r) = \frac{1}{\pi}\,e^{-r^2}\,. \tag{11.3.35}$$

is a second order cutoff. (Here we have used the subscript 2 to indicate that this cutoff is second order.) The authors then demonstrate how to obtain a fourth order cutoff by taking a linear combination of two Gaussians with different scalings

$$f_4(r) = c_1\,f_2(r) + c_2\,f_2(\frac{r}{a})\,, \tag{11.3.36}$$

where a is arbitrary subject only to $a \neq 1$. The constants c_1 and c_2 can readily be determined by noting that f_4 must satisfy (11.3.27–11.3.28) with $m=4$. For example, if we choose a such that $a^2 = 2$, then

$$f_4(r) = e^{-r^2} - \frac{1}{2}\, e^{-r^2/2}\,.$$

Similarly, one can derive sixth order cutoffs by taking a sum of the form $c_1 f_4(r) + c_2 f_4(r/a)$. A typical function of this form is

$$f_6(r) = c_1 e^{-r^2} + c_2\, e^{-r^2/2} - c_3\, e^{-r^2/4}\,.$$

Beale and Majda [1985] also constructed two dimensional cutoff functions of the form

$$f_m(r) = \frac{1}{\pi}\, P_m(r^2)\, e^{-r^2},$$

where P_m is some polynomial that must be determined. Applying (11.3.27) and the moment conditions (11.3.28) to functions of this form one finds that $P_m = Q_m - Q'_m$ where the Q_m are the Laguerre polynomials normalized so that the constant term is 1. For example,

$$\begin{aligned}
P_2(r) &= 1\,,\\
P_4(r) &= 2 - r\,,\\
P_6(r) &= 3 - 3r + \frac{r^2}{2}\,,\\
P_8(r) &= 4 - 6r + 2\,r^2 - \frac{r^3}{6}\,.
\end{aligned}$$

Both Beale and Majda [1985] and Beale [1986] have also considered high order cutoffs in three dimensions. The analogue of (11.3.35) in three dimensions is

$$f_2(r) = \frac{3}{4\pi}\, e^{-r^3}\,.$$

Beale and Majda [1985] demonstrate how to derive higher order 3D cutoffs from lower order ones by considering sums analogous to (11.3.36). They also consider sums of the form

$$f_{m+2}(r) = c_1\, f_m(r) + c_2\, r\, f'_m(r)\,.$$

Starting with a second order 3D cutoff of the form

$$f_2(r) = \frac{3}{4\pi}\left\{(1+r^6)^{-\frac{1}{2}} - r^6\,(1+r^6)^{-\frac{3}{2}}\right\},$$

Beale [1986] has used this relation to derive the following fourth and sixth order 3D cutoffs,

$$f_4(r) = \frac{3}{4\pi}\left\{(1+r^6)^{-\frac{1}{2}} + (\frac{3}{2} - r^6)\,(1+r^6)^{-\frac{3}{2}} + \frac{9r^6}{2}\,(1+r^6)^{-\frac{5}{2}}\right\},$$

$$f_6(r) = \frac{3}{4\pi}\left\{(1+r^6)^{-\frac{1}{2}} - r^6\,(1+r^6)^{-\frac{3}{2}} + \frac{27}{8}\,(1+r^6)^{-\frac{5}{2}} - \frac{135r^6}{8}\,(1+r^6)^{-\frac{7}{2}}\right\}.$$

Hald [1987] (see also Hald [1991]) has devised the following two dimensional infinite order cutoff

$$f_\infty(r) = \frac{4}{45\pi r^3}\left\{16J_3(4r) - 10J_3(2r) + J_3(r)\right\}, \qquad (11.3.37)$$

where J_3 is the Bessel function of order 3. This is one of the simplest members of a class of infinite order cutoffs based on Bessel functions which are discussed by Nordmark [1991]. This cutoff falls outside the classes of high order cutoffs studied by Anderson and Greengard [1985], Beale and Majda [1982b and 1985], and Cottet [1982]. In particular, f given by (11.3.37) fails to satisfy the moment condition (11.3.28) for $m = 4$ and conditions (11.3.29–11.3.31) for $L = m = 2$. The function simply decays too slowly. Nevertheless, by using new arguments Hald [1987] has succeeded in establishing the high order convergence of vortex methods with these cutoffs. We also remark that Hald [1987] has shown that there are no infinite order cutoffs with compact support.

Nordmark [1991] has derived the following eighth order cutoff which has compact support

$$f(r) = \begin{cases} \frac{52}{\pi}\left[1 - 21r^2 + 105r^4 - 140r^6\right]\left[1 - r^2\right]^9 & r \le 1, \\ 0 & r > 1. \end{cases} \qquad (11.3.38)$$

In his PhD thesis Nordmark [1988] conducted extensive numerical tests to compare (11.3.37) and (11.3.38). These two functions are amazingly similar. Their graphs – both in physical and Fourier space – are nearly indistinguishable from one another.

In closing we note that since f is radially symmetric in all of the above examples, then $\mathbf{K}_\delta$ can be constructed from f_δ and $\mathbf{K}$ with the aid of (11.3.34). Explicit formulas for $\mathbf{K}_\delta$ corresponding to these examples also appear in Beale and Majda [1985], Hald [1987 and 1991], and Nordmark [1991].

From a computational point of view there are several factors which must be taken into account that do not enter into the convergence theory. These are:

(i) How efficiently can $\mathbf{K}_\delta$ be computed?

(ii) Does f_δ have compact support?

The first consideration obviously determines the speed of the computation. In particular, since the cost of evaluating the velocity $\tilde{\mathbf{u}}$ given by (11.3.13) or (11.3.16) at each of the N vortex positions is $O(N^2)$, the cost of computing $\mathbf{K}_\delta$ can have a very real effect on the speed of the overall computation. A variety of techniques have been devised to reduce the $O(N^2)$ cost of the velocity computation to $O(N \log N)$ or even $O(N)$. (Some of these ideas are discussed in Section 11.7.) However these techniques usually assume that $\mathbf{K}_\delta(\mathbf{x}) = \mathbf{K}(\mathbf{x})$ for $\mathbf{x}$ sufficiently far from the center of the vortex. Hence, cutoff functions with large or unbounded support become problematic. This may also be the case when one wishes to satisfy a boundary condition of the form (11.2.1d) on $\partial\Omega$. In this regard we remark that Merriman [1991] has done some interesting work on the relation between cutoff functions and boundary conditions.

We refer the reader to the articles of Beale and Majda [1985], Hald [1991], and Nordmark [1991] for a more detailed discussion concerning the construction of high order smoothing functions. Numerical experiments to compare various cutoff functions may be found in Beale and Majda [1985], Hald and del Prete [1978], Nordmark [1991], and Perlman [1985].

11.3.5 The Convergence of the Vortex Method to Solutions of the Euler Equations

We will now state (without proof) several generic theorems which establish the validity of the vortex method approximation to solutions of the incompressible Euler equations. In the following discussion we will measure the error in both a discrete and continuous p-norm over bounded subsets of $I\!R^n$. For any function g defined on the points $\boldsymbol{\alpha}_i \in I\!R^n$, $i = 1, \ldots, N$ we let

$$\|g\|_p = \Big(\sum_i |g(\boldsymbol{\alpha}_i)|^p\, h^n\Big)^{\frac{1}{p}}.$$

Let B denote any arbitrary bounded set in $I\!R^n$. For any integrable function g defined on B we let

$$\|g\|_{L^p(B)} = \Big(\int_B |g(\mathbf{x})|^p\, d\mathbf{x}\Big)^{\frac{1}{p}}.$$

In the following discussion all statements concern the free space problem (11.2.1a–c) with boundary condition (11.2.2).

Before proceeding we remind the reader of the following. In two dimensions it has been shown that for smooth initial data solutions of the incompressible Euler equations exist and remain smooth for all time (e.g., see Kato [1967] or McGrath [1968]). In three dimensions however the existence of smooth solutions to these equations has only been established for short time (e.g., see Kato [1972]). In fact, many researchers believe that in three dimensions singularities can form in finite time, irrespective of the smoothness of the initial data. It has been hypothesized that this phenomenon is linked to the onset of turbulence. In this regard we note that Beale, Kato, and Majda [1984] have shown that if singularities do form in a solution of the 3D Euler equations, then the maximum norm of the vorticity necessarily grows without bound.

In light of the above, the most one can hope to prove in three dimensions is that the vortex method converges to the solution of the Euler equations as long as the solution remains smooth. In other words, at the present one cannot hope to prove that the numerical method converges to a solution of the 3D Euler equation for arbitrarily large times t.

The next two theorems concern the continuous time version of the vortex method. In both of these theorems one can take p to be any $1 \leq p < \infty$. The following result can, with some modifications, be found in Beale and Majda [1982b], Cottet [1982], Anderson and Greengard [1985], and Raviart [1985]. The statement as it appears here is a synopsis of these results due to Hald [1991].

Theorem 11.3.1 (Convergence in Two Dimensions) *Assume that the solution* $\mathbf{u}$ *of the incompressible Euler equations is smooth enough and that the initial vorticity field has compact support. Let* $f \in M^{L,m}$ *with* L *sufficiently large and* $m \geq 2$. *Set* $\delta = ch^q$ *with* $0 < q < 1$. *Then there exists a constant* C *such that for all* h *sufficiently small we have*

(i) convergence of particle paths

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\|_p \leq C\,[\,\delta^m + (\frac{h}{\delta})^L \delta\,]\,,$$

(ii) convergence of particle velocities

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{u}}(\tilde{\mathbf{x}}_i(t), t) - \mathbf{u}(\mathbf{x}_i(t), t)\|_p \leq C\,[\,\delta^m + (\frac{h}{\delta})^L \delta\,]\,,$$

(iii) convergence of the velocity field

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{u}}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t)\|_{L^p(B)} \leq C\,[\,\delta^m + (\frac{h}{\delta})^L \delta\,]\,.$$

Note that in the above theorem we have not stated a result analogous (ii) and (iii) for the vorticity field. Proofs that the approximate vorticity field converges to the exact vorticity field in two dimensions do appear in the literature however. For example, Hald and Del Prete [1978] prove a statement analogous to (iii) for the vorticity in the maximum norm for short times and for a specific choice of cutoff function. The general understanding of the theory as well as the analytical tools available have improved considerably since their proof. In fact, the current state of the theory is such that, if everything is smooth enough, then obtaining bounds for the vorticity from statements of the form (i–iii) above should be relatively straightforward. However, most workers do not bother to explicitly state and prove such results. In particular, we know of no explicit statement in the literature analogous to (ii–iii) for the vorticity field in two dimensions.

In three dimensions there is a notable exception to this trend. The following theorem is due to Beale [1986]. (See also Cottet [1988b].) What is meant by $\tilde{\omega}$ here is the function defined by (11.3.24)

$$\tilde{\omega}(\mathbf{x}, t) = \sum_i \zeta_\delta(\mathbf{x} - \mathbf{x}_i(t))\,\omega_i(t)\,h^3\,.$$

Theorem 11.3.2 (Convergence in Three Dimensions) *Assume that the solution* $\mathbf{u}$ *of the incompressible Euler equations exists and and remains smooth on the interval* $0 \leq t \leq T$ *and that the initial vorticity field has compact support. Let* $f \in M^{L,m}$ *with* L *sufficiently large and* $m \geq 4$.[4] *Set* $\delta = ch^q$ *with* $\frac{1}{3} < q < 1$. *Then there exists a constant* C *such that for all* h *sufficiently small we have*

(i) convergence of particle paths

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\|_p \leq Ch^{mq} ,$$

(ii) convergence of particle velocities

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{u}}(\tilde{\mathbf{x}}_i(t), t) - \mathbf{u}(\mathbf{x}_i(t), t)\|_p \leq Ch^{mq} ,$$

(iii) convergence of the velocity field

$$\max_{0 \leq t \leq T} \|\tilde{\mathbf{u}}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t)\|_{L^p(B)} \leq Ch^{mq} ,$$

(iv) convergence of particle vorticities

$$\max_{0 \leq t \leq T} \|\tilde{\boldsymbol{\omega}}(\tilde{\mathbf{x}}_i(t), t) - \boldsymbol{\omega}(\mathbf{x}_i(t), t)\|_p \leq Ch^{mq-1},$$

(v) convergence of the vorticity field

$$\max_{0 \leq t \leq T} \|\tilde{\boldsymbol{\omega}}(\mathbf{x}, t) - \boldsymbol{\omega}(\mathbf{x}, t)\|_{L^p(B)} \leq Ch^{mq-1} .$$

It is apparent from these two theorems that an essential requirement of the theory is that

$$\frac{h}{\delta} \to 0 , \qquad \text{as} \quad h , \delta \to 0 .$$

In other words, the mesh length h must go to zero faster than the cutoff parameter δ. This implies that the vortex cores tend to overlap at an increasing rate as h, $\delta \to 0$.

Since $\delta = ch^q$, in Theorem 11.3.2 we see that the error in (i–iii) is of order δ^m. This is the motivation for calling $f \in M^{L,m}$ an mth order cutoff. If in Theorem 11.3.1 the constant L is such that

$$L > \frac{(m-1)q}{(1-q)} , \tag{11.3.39}$$

then it is easy to see that the convergence rate is also $O(h^{mq}) = O(\delta^m)$. If we choose q close to 1, then such errors are essentially of order h^m. However, by (11.3.39) this

[4] Cottet [1988b] has shown that it is sufficient to take $m \geq 2$.

means that $L \sim (1-q)^{-1}$ must be very large. Since one of the assumptions (which we have not stated) in both of these theorems is that the flow field is at least C^{L+n+1}, this will only work for very smooth flow fields.

From a computational point of view it is evident that choosing q close to 1 in Theorem 2 yields the fastest rate of convergence. However this is based on the assumption that the underlying flow field is very smooth. Roughly speaking, for $f \in M^{L,m}$ if the flow is at least C^{L+n+1}, then the optimal choice for δ will be $\delta = h^{1-\epsilon}$ where $\epsilon \ll 1$ (e.g., see Beale and Majda [1982b], Cottet [1982 and 1991], Anderson and Greengard [1985] and Raviart [1985]). On the other hand, if the flow has only a small number of derivatives and the cutoff is of high order, then choosing $\delta = h^{1/2}$ seems to be best (e.g., see Hald [1979]). In particular, this holds for infinite order cutoffs (e.g., see Hald [1987]).

In practice one usually takes h to be as small as possible and then chooses δ based on heuristic considerations. For an example: How does the solution look? How does the method perform on a simple test problem with a given choice of parameters? For example, see Sethian and Ghoniem's [1988] comprehensive study of the flow past a backward facing step. We also remark that numerical experiments appear to indicate that a larger δ yields a more accurate solution at late times. We refer the interested reader to Beale, Eydeland, and Turkington [1991], Beale and Majda [1985], Hald and Del Prete [1978], Nordmark [1991], and Perlman [1985] for a more detailed discussion of these issues.

Convergence of the Time Discretized Method In practice a vortex method computation also involves the discretization of (11.3.6a,b) or (11.3.19a,b)–(11.3.20a,b) in time. Anderson and Greengard [1985] were the first to prove that the time discretized vortex method converges to solutions of the incompressible Euler equations. They studied the modified Euler method and an explicit class of multi-step schemes that includes the Adams-Bashforth methods. Their proof for the modified Euler method can be extended to include other second order Runge-Kutta methods. The treatment of higher-order Runge-Kutta methods is more difficult. Hald [1987] has proved the convergence of the classical fourth order Runge-Kutta method for vortex methods with infinite order cutoffs. The following theorem is a synopsis of Anderson and Greengard's result.

Theorem 11.3.3 (Convergence of the Time Discretized Vortex Method) *Suppose that the hypotheses of Theorem 11.3.1 (respectively Theorem 11.3.2) hold for $n = 2$ (respectively $n = 3$) and that the fluid flow map $\mathbf{x} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is $s+1$ times continuously differentiable. Assume further that the ODE solver is a member of the class of multi-step methods considered by Anderson and Greengard [1985] and that it has a local truncation error which is order Δt^{r+1}. Let $T = k_0 \Delta t$ and for $0 \leq k \leq k_0$ let $\mathbf{x}_i^k = \mathbf{x}_i(k\Delta t)$. Then* *there exists a constant C such that for all h, Δt sufficiently small we have*

(i) convergence of particle paths

$$\max_{0\le k\le k_0} \|\tilde{\mathbf{x}}_i^k - \mathbf{x}_i^k\|_2 \le C\,[\,\delta^m + (\frac{h}{\delta})^L\delta + h^s + \Delta t^r\,]\,. \qquad (11.3.40)$$

11.3.6 The Point Vortex Method

If we let f_δ be the Dirac delta function, $f_\delta(\mathbf{x}) = \delta(\mathbf{x})$, then in $I\!R^2$ the approximate vorticity (11.3.9) becomes a collection of point masses

$$\tilde{\omega}(\mathbf{x},t) = \sum_i \delta(\mathbf{x} - \mathbf{x}_i(t))\,\omega_i(t)\,h^2\,,$$

and the approximate velocity becomes

$$\tilde{\mathbf{u}}(\mathbf{x},t) = \sum_i \mathbf{K}(\mathbf{x} - \mathbf{x}_i(t))\omega_i(t)h^2\,.$$

This is the "classical" *point vortex method.* As we have already noted, the singularity in $\mathbf{K}$ gives rise to an arbitrarily large velocity as $\mathbf{x} \to \mathbf{x}_i$. For this reason it was generally believed that the point vortex method was unstable and hence would not converge to solutions of the incompressible Euler equations. However, Goodman, Hou, and Lowengrub [1990a] have recently proved that the point vortex method does in fact converge to solutions of two dimensional Euler equations. The key idea in their proof is that for smooth flow the smoothness of the flow map and its inverse guarantee that for finite times two particles will remain sufficiently far apart. In fact, for any finite time two neighboring particles which are initially separated by a distance h will remain separated by a distance which is $O(h)$. Thus, the point vortex method in essence has a cutoff which is asymptotically $O(h)$ for any finite time. The authors also prove that 2D point vortex method is second order. This follows from the interesting fact that certain errors in the stability estimate cancel because the Biot-Savart kernel $\mathbf{K}$ is odd. (Beale [1986] used this fact to improve his accuracy estimates for a 3D vortex blob method.)

Hou and his colleagues have also been able to prove that three dimensional point vortex methods converge to solutions of the Euler equations for sufficiently smooth flows. It turns out that the analytical tools one needs to prove this depend on how one discretizes the stretching term (11.3.15). Hou and Lowengrub [1990] have used an argument similar to the one used in two dimensions to prove the convergence of a method in which the stretching term is approximated by a Lagrangian finite differencing such as in equation (11.3.21). When the stretching term is "grid-free" as in (11.3.20a,b) the analysis is somewhat harder. Cottet, Goodman and Hou [1991] have used several additional ideas to establish the convergence of a three dimensional point vortex method of this type.

As Hou [1991] has pointed out, the convergence analysis does not necessarily imply that the point vortex method is practical without smoothing or some form of desingularization. The theory states that for any given time T there is a constant $h_0(T)$ such that for all $h \leq h_0$ the method is stable and converges. However, h_0 may be impractically small. Furthermore, given a fixed discretization of the initial vorticity field, there will be a critical time t_c beyond which two neighboring particles may become so close that the stability analysis no longer applies.

These observations have lead Hou and his co-workers to consider several possible remedies. Hou [1992] has proposed a method for "desingularizing" the integral kernel $\mathbf{K}$ that seems to work well. Hou, Lowengrub, and Shelley [1991] have studied a regridding technique for the desingularized point vortex method. The basic idea behind a regridding algorithm is to stop the computation when the error begins to rise and reinitialize the vortex positions and strengths in a manner that will maintain a given level of accuracy. Regridding algorithms and other techniques for improving the long time accuracy of vortex methods have also been studied by Beale [1988] and Nordmark [1991]. Hald [1991] has reviewed some of these efforts.

For further discussion concerning the point vortex method and related topics see Hou [1991].

11.4 The No-Flow Boundary Condition

We now discuss the modifications to the computed flow field that are required in order to satisfy the no-flow boundary condition (11.2.1d) on $\partial\Omega$. Assume that we have a solution $\mathbf{u}_\omega$ of (11.2.1a–c) that fails to satisfy (11.2.1d) on $\partial\Omega$. We seek a scalar function ϕ such that

$$\Delta\phi = 0 \qquad \text{in} \quad \Omega\,, \tag{11.4.1a}$$

$$\nabla\phi \cdot \boldsymbol{\eta} = -\mathbf{u}_\omega \cdot \boldsymbol{\eta}, \qquad \text{on} \quad \partial\Omega\,. \tag{11.4.1b}$$

If we define $\mathbf{u}_p \equiv \nabla\phi$ then, as we have shown in Section 11.2.1, the velocity field $\mathbf{u} \equiv \mathbf{u}_\omega + \mathbf{u}_p$ satisfies (11.2.1a–d). The velocity $\mathbf{u}_p$ is called a *potential flow* (e.g., see Chorin and Marsden [1990]).

Numerically we attempt to perform a discrete analog of the continuous solution just described. In general, at the end of each time step we have some function $\mathbf{u}_\omega$ which fails to satisfy (11.2.1d). The problem is to find ϕ such that ϕ satisfies (11.4.1a,b). In other words we wish to solve Laplace's equation for ϕ subject to Neumann boundary conditions. In some instances this can be done exactly by using the method of images or some similar idea. In other instances it is necessary to determine a numerical approximation $\tilde{\phi}$ to ϕ. In either case, effective numerical techniques for determining $\mathbf{u}_p$ or an approximation $\tilde{\mathbf{u}}_p$ may often be found in the scientific literature. We describe several simple examples here.

To begin, note that the solution of the potential flow problem is dictated in large part by the shape of the domain. For example, let $n = 2$ and suppose that Ω is the upper half plane $\{(x, y) : y > 0\}$. If $\mathbf{u}_\omega$ is the velocity due to a collection of N vortices at (x_i, y_i) with strengths ω_i, then it is a simple matter to check that the potential flow $\mathbf{u}_p$ we seek is precisely the flow due to a collection of N vortices at $(x_i, -y_i)$ with strengths $-\omega_i$. Here we have employed the well known *method of images* (e.g., see Garabedian [1986, pp. 246–247]) to find a solution of (11.4.1a,b) in the half plane $y > 0$.

As another example consider flow past a circle of radius R centered at the origin in two dimensions. So $\Omega = \{\mathbf{x} : |\mathbf{x}| > R\}$ and $\partial\Omega = \{\mathbf{x} : |\mathbf{x}| = R\}$. Suppose $\mathbf{u}_\omega$ is the flow induced by a collection of vortices at $\mathbf{x}_i \in \Omega$ with strengths ω_i, $i = 1, \ldots, N$. Let

$$\mathbf{x}_i' = \frac{R^2}{|\mathbf{x}_i|^2}\,\mathbf{x}_i$$

be the radial image of the point $\mathbf{x}_i$ with respect to the circle $\partial\Omega$. Then a potential flow $\mathbf{u}_p$ which cancels the normal component of $\mathbf{u}_\omega$ on $\partial\Omega$ is the flow due to a collection of vortices with positions $\mathbf{x}_i'$ and strengths $-\omega_i$, $i = 1, \ldots, N$, (e.g., see Garabedian [1986, pp. 247–249]). However, the total circulation about $\partial\Omega$,

$$\Gamma_\Omega = \oint_C \mathbf{u}_\omega \cdot \mathbf{ds}\,, \tag{11.4.2}$$

has now changed. Here C is any circle of radius $r > \max|\mathbf{x}_i|$ centered at the origin. Another potential flow that satisfies

$$\mathbf{u}_p \cdot \boldsymbol{\eta} = -\mathbf{u}_\omega \cdot \boldsymbol{\eta}$$

and yet leaves (11.4.2) unchanged is the flow due to the N radial images at positions $\mathbf{x}_i'$ and strengths $-\omega_i$ together with N additional vortices, all located at the origin, and having strengths ω_i.

This illustrates an important fact about the potential flow. Sometimes it is necessary to consider other properties of the flow field, such as total circulation, when choosing $\mathbf{u}_p$. Also note that, in general, ϕ is not unique. Given any solution ϕ of (11.4.1a,b), the function $\phi + c$ for any constant c, is also a solution of (11.4.1a,b).

For more general domains the method of images is not always practical, or necessarily even possible. In these cases one can resort to one of several different strategies. Sometimes it is possible to conformally map Ω onto another domain Ω' for which the method of images will work. However there are potential pitfalls with this approach.

Large numerical errors can occur as a result of the conformal mapping. For example, this might occur in regions where $\partial\Omega$ is not differentiable, such as at a corner. Accurate numerical algorithms for conformal mapping are an active area of research and results in this area will ultimately impact vortex methods (e.g., see Howell [1990] and Howell and Trefethen [1990].)

Another approach to solving the potential flow problem is to place a grid over the computational domain and use a fast Poisson solver on the grid. In this case one must be careful to ensure that the introduction of the grid does not also introduce the very types of diffusive errors that the vortex method was designed to eliminate. For example, see the method of Mayo [1985]. We will discuss this issue further in Section 11.7. One can also use a multigrid method or a finite element method to solve the potential flow problem (11.4.1a,b).

Finally we mention that there have been a number of recent advances in the fast numerical solution of integral equations. Thus an effective way to solve (11.4.1a,b) might be to reformulate (11.4.1a,b) as an integral equation and use a fast solver to obtain an approximation to the potential flow $\mathbf{u}_p$. These techniques have the advantage of being applicable to a wide variety of domains. We refer interested reader to Greenbaum, Greengard and McFadden [1991], Greengard [1988], and Rokhlin [1985] for further details.

11.5 Vortex Method Approximations to Solutions of the Navier-Stokes Equations

We now consider vortex methods for computing viscous flow. We begin by describing the granddaddy of all such methods, the *random vortex method*. As its name connotes, the random vortex method is essentially a vortex method with the addition of a random approximation to the viscous terms in the Navier-Stokes equations. After describing this method we state several theorems that have been proved concerning its convergence to solutions of the Navier-Stokes equations. We then describe several vortex methods in which the random walk has been replaced by a non-random approximation to the viscous terms in the Navier-Stokes equations. We also state a convergence theorem that has been proved for one of these methods. We close with a few remarks regarding random versus non-random vortex methods and a brief discussion of some of the important open questions in the field.

11.5.1 The Random Vortex Method

The random vortex method was first introduced by Chorin [1973]. We will now describe this method as it is used to approximate solutions of the Navier-Stokes equations (11.2.14a–d) in $\Omega = I\!R^n$ with the free space boundary condition (11.2.2). In Section

6 we will discuss the modifications to this algorithm that are required when Ω is a bounded domain with boundary $\partial\Omega$ upon which the boundary condition (11.2.14e) must be satisfied.

In practice one is usually interested in determining the velocity field $\tilde{\mathbf{u}}^k = \mathbf{K} * \tilde{\omega}^k$ at some time $t_k = k\Delta t$ rather than in just determining the corresponding vorticity field $\tilde{\omega}^k$. Most theoretical work on the random vortex method has been concerned with estimating the error between the approximate and exact velocity field, rather than the error between the corresponding vorticity fields. For example, it has been proved that this velocity field converges to the exact velocity field in the appropriate limit. (In fact, at the time of this writing, no analogous statement concerning the vorticity field has yet been proved.) However, since the pressure term is eliminated from the momentum equations when they are written in vorticity form, we prefer to introduce the random vortex method as a fractional step method for solving (11.2.14a–d) for the approximate vorticity field $\tilde{\omega}^k$.

Let

$$\tilde{\omega}^k(\mathbf{x}) = \sum_{j=1}^{N} f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j^k)\,\boldsymbol{\Gamma}_j^k\,, \tag{11.5.1}$$

where it is understood that $\boldsymbol{\Gamma}_j^k = \omega_j h^2$ when $n = 2$ and $\boldsymbol{\Gamma}_j^k = \boldsymbol{\omega}_j^k h^3$ when $n = 3$. We wish to emphasize that when $n = 3$ one can just as well replace (11.5.1) with (11.3.24); the details of the computation remain the same. The random vortex method is a fractional step method. The first step consists of approximating the solution of the Euler equations (11.2.13a–c) at time $t_{k+1} = t_k + \Delta t$ with initial data (11.5.1) given at time t_k. This is accomplished by updating the vortex positions $\tilde{\mathbf{x}}_j^k$ and strengths $\boldsymbol{\Gamma}_j^k$ as described in Sections 11.3.1 and 11.3.2. This yields an intermediate vorticity field of the form,

$$\tilde{\omega}^{k+\frac{1}{2}}(\mathbf{x}) = \sum_j f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j^{k+\frac{1}{2}})\,\boldsymbol{\Gamma}_j^{k+1}\,. \tag{11.5.2}$$

Here $\tilde{\mathbf{x}}_j^{k+\frac{1}{2}}$ simply denotes the particle positions at the end of the first of two steps; i.e., as a result of (11.3.14) or perhaps some higher order ODE solver. If $n = 2$, then $\boldsymbol{\Gamma}_j^{k+1} = \boldsymbol{\Gamma}_j^k$. Otherwise, we assume that $\boldsymbol{\Gamma}_j^{k+1} = \boldsymbol{\omega}_j^{k+1} h^3$ where $\boldsymbol{\omega}_j^{k+1}$ is given by (11.3.22) or some higher order time discretization of (11.3.20a).

The second fractional step is an approximate solution of the heat equation,

$$\omega_t = \nu\Delta\omega\,, \tag{11.5.3}$$

at time t_{k+1} with initial data (11.5.2) at time t_k. This is accomplished by letting each vortex undergo a random walk,

$$\tilde{\mathbf{x}}_j^{k+1} = \tilde{\mathbf{x}}_j^{k+\frac{1}{2}} + \boldsymbol{\eta}_j\,,$$

where $\boldsymbol{\eta}_j = (\eta_j^1, \ldots, \eta_j^n)$ and the $\{\eta_j^i : i = 1, \ldots, n,\ j = 1, \ldots, N\}$ are independent, Gaussian distributed random numbers with mean 0 and variance $2\nu\Delta t$. The approximate vorticity field at time t_{k+1} is therefore,

$$\tilde{\boldsymbol{\omega}}^{k+1}(\mathbf{x}) = \sum_j f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j^{k+1})\, \boldsymbol{\Gamma}_j^{k+1} .$$

Why random walk the vortices? Suppose one wishes to solve the diffusion equation (11.5.3) in $\Omega = I\!R^n$ with initial data (11.5.2). The exact solution at time t is given by,

$$\begin{aligned}\boldsymbol{\omega}(\mathbf{x}, t) &= (4\pi\nu t)^{-n/2} \int_{I\!R^n} e^{-(\mathbf{x}-\boldsymbol{\eta})^2/4\nu t}\, \tilde{\boldsymbol{\omega}}^{k+\frac{1}{2}}(\boldsymbol{\eta})\, d\boldsymbol{\eta} \\ &= (4\pi\nu t)^{-n/2} \int_{I\!R^n} \Big(\sum_j f_\delta(\mathbf{x} - (\tilde{\mathbf{x}}_j^{k+\frac{1}{2}} + \boldsymbol{\eta}))\, \boldsymbol{\Gamma}_j^{k+1} \Big)\, e^{-\boldsymbol{\eta}^2/4\nu t}\, d\boldsymbol{\eta} \\ &= E\Big[\sum_j f_\delta(\mathbf{x} - (\tilde{\mathbf{x}}_j^{k+\frac{1}{2}} + \boldsymbol{\eta}))\, \boldsymbol{\Gamma}_j^{k+1} \Big] ,\end{aligned}$$

where E denotes expectation over Gaussian random variables η on $I\!R^n$ with mean 0 and variance $2\nu t$. Thus, the exact solution at time t_{k+1} of the diffusion equation (11.5.3) with initial data (11.5.2) at time t_k is *precisely* the expected value of the function obtained from (11.5.2) by letting each of the vortex positions $\tilde{\mathbf{x}}_j^{k+\frac{1}{2}}$ undergo an independent random walk with mean 0 and variance $2\nu\Delta t$.

Millanazo and Saffman [1977] and Roberts [1985] have studied the accuracy of the random vortex method in two dimensions by using it to approximate a known radially symmetric solution of the Navier-Stokes equations. These studies indicate that the rate of convergence is $O(\sqrt{\nu/N})$ where N is the number of vortices and ν is the kinematic viscosity. In $I\!R^2$ this corresponds to a rate which is first order in h. To see this suppose that at time $t = 0$ the support of the vorticity $\Omega_0 \subset I\!R^2$ is covered by N equal cells of side $h = \overline{\sqrt{N}}^{-1}$. Then the observed convergence rate for the random vortex method is $O(\overline{\sqrt{N}}^{-1}) = O(h)$. Similarly, one can show that in $I\!R^3$ a rate that is $O(\overline{\sqrt{N}}^{-1})$ corresponds to a rate that is $O(h^{3/2})$.

11.5.2 More on Initial Conditions

In practice the random vortex method tends to be used in one of two ways. One way is to prescribe an initial vorticity field $\boldsymbol{\omega}^0$ which is compactly supported in some unbounded domain such as $\Omega = I\!R^n$. In this case the initial vortex positions and weights may be determined as described in Section 11.3.3. However, in many engineering applications no initial vorticity field is specified. The initial condition often consists of a bounded domain Ω in which a potential flow that satisfies the no-flow boundary condition on $\partial\Omega$ is given. As the computation progresses vorticity is allowed to enter the flow via some vortex creation mechanism designed to mimic the physical creation of vorticity at $\partial\Omega$. We discuss techniques for doing this in Section 11.6. At the current time there is no complete theory for the convergence of such a computation to the exact solutions of the Navier-Stokes equations.

11.5.3 Convergence of the Random Vortex Method

Since its introduction there has been considerable interest in proving that the random vortex method converges to solutions of the Navier-Stokes equations. However, it has not been until the last decade that this fact has been rigorously established. We will now briefly review what has been proved to date about the random vortex method.

Marchiorio and Pulverenti [1982] were the first establish a convergence theorem for the random vortex method. They showed that solutions of the stochastic differential equation which is the continuous time version of the random vortex method converge to solutions of the Navier-Stokes equations in $\mathbb{R}^2$. However, their result does not exhibit a rate of convergence in terms of the parameters h, δ, Δt, and ν which is desirable from a computational point of view.

Subsequently, Goodman [1985] proved that a time discretized version of the random vortex method converges to solutions of the Navier-Stokes equations in $\mathbb{R}^2$. In particular, Goodman was able to show that for any fixed $t_k = k\Delta t$,

$$\sup_{\mathbf{x}\in B} |\tilde{\mathbf{u}}^k(\mathbf{x}) - \mathbf{u}(\mathbf{x}, t_k)| \leq C N^{-\frac{1}{4}} \log N$$

with high probability. Here B is an arbitrary bounded set in $\mathbb{R}^2$. Thus, Goodman established a convergence rate which is $O(N^{-\frac{1}{4}} \log N)$. This is essentially one half order slower than the rate of $N^{-\frac{1}{2}}$ which one might expect based on the work of Millanazo and Saffman [1977] and Roberts [1985] that was mentioned above.

Long [1988 and 1990] has been able to establish a faster rate of convergence for the *time continuous* version of the random vortex method in both two and three dimensions. We state a synopsis of Long's results here (see also Hald [1991]). In the following it is assumed that the initial vorticity field has support contained in a bounded set $\Omega_0 \subset \mathbb{R}^n$ and that the initial discretization of the vorticity field is as described in Section 11.3.4. In particular, we assume that Ω_0 has been covered with a square grid of side $h = O(N^{-1/n})$, for $n = 2$ or 3, that the initial vortex positions $\tilde{\mathbf{x}}_j(0)$ are taken to be the centers of the grid points, and that the initial vortex strengths are of the form $\Gamma_j(0) = \omega^0(\tilde{\mathbf{x}}_j(0))\, h^n$. In both of the following theorems p is any real number that satisfies $1 \leq p < \infty$.

Theorem 11.5.1 (Convergence in Two Dimensions) *Suppose that ω^0 is supported inside a bounded set $\Omega_0 \subset I\!R^2$ and that ω^0 smooth enough that the solution ω of the Navier-Stokes equations (11.2.14a–d) with initial data ω^0 and boundary condition (11.2.2) exists and is sufficiently smooth on the interval $0 \le t \le T$. Let $\mathbf{u} = \mathbf{K} * \omega$ be the associated velocity field. Let $f \in M^{L,m}$ with L sufficiently large and $m \ge 2$. Set $\delta = ch^q$ with $0 < q < 1$ and let B be a bounded set in $I\!R^2$. Then there exists a constant C such that for all h sufficiently small we have*

(i) convergence of the particle paths

$$\max_{0 \le t \le T} \|\tilde{\mathbf{x}}_j(t) - \mathbf{x}_j(t)\|_p \le C[\delta^m + (\frac{h}{\delta})^L \delta + h|\log h|]$$

(ii) convergence of the particle velocities

$$\max_{0 \le t \le T} \|\tilde{\mathbf{u}}(\tilde{\mathbf{x}}_j(t), t) - \mathbf{u}(\mathbf{x}_j(t), t)\|_p \le C\ [\delta^m + (\frac{h}{\delta})^L \delta + h|\log h|]$$

(iii) convergence of the velocity field

$$\max_{0 \le t \le T} \|\tilde{\mathbf{u}}(\mathbf{x}(t), t) - \mathbf{u}(\mathbf{x}, t)\|_{L^p(B)} \le C[\delta^m + (\frac{h}{\delta})^L \delta + h|\log h|]$$

with probability $1 - o(h)$.

Note that the computed velocity field $\tilde{\mathbf{u}}$ is a random quantity, whereas the velocity $\mathbf{u}$ is a deterministic quantity. This theorem says that the error between them is small with high probability. In other words, if we run the problem just once with h sufficiently small, then we should expect an accurate solution.

The proof of this theorem may be found in Long [1988]. The statement as it appears here is a synopsis due to Hald [1991]. Long's convergence rate is nearly optimal in the sense that for q close to 1 (i.e. δ close to h) the dominant term in the error is,

$$O(h\,|\log h|) = O(\sqrt{N}^{-1} \log N) \approx O(\sqrt{N}^{-1}).$$

We note that in the absence of some variance reduction technique one cannot hope to obtain a rate that is better than $O(\sqrt{N}^{-1})$. (See Handscomb and Hammersley [1964] or Maltz and Hitzl [1979] for a justification of this statement and a discussion of variance reduction techniques.)

Long has also established the convergence of the random vortex method in $I\!R^3$.

Theorem 11.5.2 (Convergence in Three Dimensions) *Suppose that ω^0 is supported inside a bounded set $\Omega_0 \subset I\!\!R^3$ and that ω^0 smooth enough that the solution ω of the Navier-Stokes equations (11.2.14a–d) with initial data ω^0 and boundary condition (11.2.2) exists and is sufficiently smooth on the interval $0 \le t \le T$. Let $\mathbf{u} = \mathbf{K} * \omega$ be the associated velocity field. Let $f \in M^{L,m}$ with L sufficiently large and $m \ge 4$. Set $\delta = ch^q$ with $0 < q < \frac{3}{5}$. Let B be a bounded set in $I\!\!R^3$. Then there exists a constant C such that for all h sufficiently small we have*

(i) convergence of the particle paths

$$\max_{0 \le t \le T} \|\tilde{\mathbf{x}}_j(t) - \mathbf{x}_j(t)\|_p \le C\left[\,\delta^m + h\,(\frac{h}{\delta})^{1/2}\,|\log h|\,\right],$$

(ii) convergence of the particle velocities

$$\max_{0 \le t \le T} \|\tilde{\mathbf{u}}(\tilde{\mathbf{x}}_j(t), t) - \mathbf{u}(\mathbf{x}_j(t), t)\|_p \le C\left[\,\delta^m + h\,(\frac{h}{\delta})^{1/2}\,|\log h|\,\right],$$

(iii) convergence of the velocity field

$$\max_{0 \le t \le T} \|\tilde{\mathbf{u}}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}, t)\|_{L^p(B)} \le C\left[\,\delta^m + h\,(\frac{h}{\delta})^{1/2}\,|\log h|\,\right],$$

with probability $1 - o(h)$.

The proof of this theorem may be found in Long [1990]. To the author's knowledge at the present time it is the only convergence proof for the random vortex method in $I\!\!R^3$.

In both of the above theorems the constant C depends only on T, k, m, p, q, the diameter of B, the diameter of Ω_0 and the bounds for a finite number of derivatives of the velocity field $\mathbf{u}(\mathbf{x}, t)$. For simplicity we have actually stated a somewhat weaker form of Long's results. Long [1988 and 1990] can replace the $1 - o(h)$ probability of having a high error by a term of the form $1 - o(h^\gamma)$ where the constant γ can be chosen to be as large as one likes. However, the constant C on the right hand sides of the error bounds depends on γ and, in general, will increase as γ increases.

At the present time all convergence theorems for the random vortex method are for $\Omega = I\!\!R^n$ with the free space boundary condition (11.2.2). The convergence of the method to solutions of the Navier-Stokes equations in bounded domains $\Omega = I\!\!R^n$ with the boundary condition (11.2.14e) on $\partial\Omega$ is still an open and difficult problem. We will discuss these issues further in Section 11.6.

11.5.4 Deterministic Vortex Methods

During the past decade much research has been devoted to improving the convergence rate of the random vortex method. For example, Chang [1988] has applied high order numerical techniques that he developed for the integration of stochastic differential equations (Chang [1987]) to a random vortex method computation. However the vast majority of these efforts have been directed at finding a non-random technique for replacing the random walk algorithm with a higher order approximation to solutions of the diffusion equation (11.5.3). These methods are generically called *deterministic vortex methods*. This is a very active area of research and there are still many open questions in the field. We will now briefly illustrate some ideas that are currently being considered.

A very natural and perhaps one of the first examples of a deterministic vortex method is what has become known as the *core spreading algorithm*. (See Kuwahara and Takami [1973] or Leonard [1980] for details.) In the two dimensional (continuous time) version of this algorithm the approximate vorticity has the form

$$\tilde{\omega}(\mathbf{x},t) = \sum_j f_t(\mathbf{x} - \tilde{\mathbf{x}}_j(t))\, \Gamma_j\,, \tag{11.5.4}$$

where the core function f_t now depends on the time t rather than on a constant cutoff parameter δ. The function $f_t(\mathbf{x})$ is chosen to be the solution at time t of the heat equation with initial data f_0 at time $t = 0$,

$$f_t(\mathbf{x}) = (G_t * f_0)(\mathbf{x}) = \frac{1}{4\pi\nu t} \int e^{-(\mathbf{x}-\mathbf{y})^2/4\nu t} f_0(\mathbf{y})\, d\mathbf{y}\,.$$

Here G_t denotes the heat kernel in $I\!R^2$,

$$G_t(\mathbf{x}) = \frac{1}{4\pi\nu t}\, e^{-\mathbf{x}^2/4\nu t}\,. \tag{11.5.5}$$

If one chooses f_0 to be the Dirac delta function $f_0(\mathbf{x}) = \delta(\mathbf{x})$, then

$$f_t(\mathbf{x}) = \frac{1}{4\pi\nu t} e^{-\mathbf{x}^2/4\nu t}\,.$$

In other words, f_t is a Gaussian smoothing function but now the "mass" of the core spreads out in time at a rate proportional to $\sqrt{\nu t}$. In general, f_0 can be taken to be any one of the two dimensional smoothing functions discussed in Section 11.3.4.

If we let $\mathbf{K}_t = \mathbf{K} * f_t$, then the approximate vorticity (11.5.4) yields an approximate velocity field of the form,

$$\tilde{\mathbf{u}}(\mathbf{x},t) = \sum_j \mathbf{K}_t(\mathbf{x} - \tilde{\mathbf{x}}_j(t))\, \Gamma_j\,. \tag{11.5.6}$$

One might expect the velocity field $\tilde{\mathbf{u}}$ to converge to a solution $\mathbf{u}$ of the Navier-Stokes equations. However Greengard [1985] has shown that it converges to a solution of the wrong equation! We briefly outline his argument here.

Let $\mathbf{X}(\alpha, t)$ denote the solution of the Lagrangian equations

$$\frac{d\mathbf{X}}{dt}(\boldsymbol{\alpha}, t) = \int \mathbf{K}_t(\mathbf{X}(\boldsymbol{\alpha}, t) - \mathbf{X}(\boldsymbol{\alpha}', t))\, \omega^0(\boldsymbol{\alpha}')\, d\boldsymbol{\alpha}' \,, \tag{11.5.7a}$$

$$\mathbf{X}(\boldsymbol{\alpha}, 0) = \boldsymbol{\alpha} \,. \tag{11.5.7b}$$

where ω^0 is the initial vorticity field. (Compare this with equations (11.3.4a,b).) Let $\mathbf{U}$ denote the velocity field corresponding to (11.5.7a,b),

$$\mathbf{U}(\mathbf{x}, t) = \int \mathbf{K}_t(\mathbf{x} - \mathbf{X}(\boldsymbol{\alpha}, t))\, \omega^0(\boldsymbol{\alpha})\, d\boldsymbol{\alpha} \,. \tag{11.5.8}$$

So (11.5.7a) can be rewritten as $(d\mathbf{X}/dt)(\boldsymbol{\alpha}, t) = \mathbf{U}(\mathbf{X}(\boldsymbol{\alpha}, t), t)$. Greengard first showed that for appropriately chosen f_0 solutions of

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \sum_j \mathbf{K}_t(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \Gamma_j \,, \tag{11.5.9a}$$

$$\tilde{\mathbf{x}}_i(0) = \boldsymbol{\alpha}_i \,. \tag{11.5.9b}$$

converge to solutions of (11.5.7a,b). In other words he showed that $\tilde{\mathbf{u}} \to \mathbf{U}$ where $\tilde{\mathbf{u}}$ is given by (11.5.6) and $\mathbf{U}$ is given by (11.5.8).

Thus, it suffices to compare $\mathbf{U}$ to solutions $\mathbf{u}$ of the Navier-Stokes equations. Let $\xi = \nabla \times \mathbf{U}$ denote the vorticity field corresponding to $\mathbf{U}$. We compare ξ to solutions $\omega = \nabla \times \mathbf{u}$ of (11.2.14a–d) with initial data ω^0 and boundary condition (11.2.2). Let ζ denote the passive transport of ω^0 in the velocity field $\mathbf{U}$. In other words, $\zeta(\mathbf{X}(\boldsymbol{\alpha}, t), t) = \omega^0(\boldsymbol{\alpha})$ and hence,

$$\partial_t \zeta + \mathbf{U} \cdot \nabla \zeta = 0 \,.$$

Changing variables $\boldsymbol{\alpha} \to \mathbf{X}(\boldsymbol{\alpha}, t)$ in (11.5.8) yields $\mathbf{U} = \mathbf{K}_t * \zeta = \mathbf{K} * (G_t * \zeta)$. Hence $\xi = \nabla \times \mathbf{U} = G_t * \zeta$.

Now the exact vorticity ω satisfies

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega \,. \tag{11.5.10}$$

But if we differentiate $\xi = G_t * \zeta$ with respect to t and use the fact that $\partial_t (G_t * \zeta) = \nu \Delta (G_t * \zeta) + G_t * (\partial_t \zeta)$ we find that ξ satisfies

$$\frac{\partial \xi}{\partial t} + G_t * (\mathbf{U} \cdot \nabla \zeta) = \nu \Delta \xi \,. \tag{11.5.11}$$

By differentiating (11.5.10) and (11.5.11) once again with respect to t one can show that the difference $\omega_t - \xi_t$ satisfies an ODE in t which is not identically zero unless the initial data ω^0 is radially symmetric. Hence, the exact solution ω of (11.5.10) with initial data ω^0 is not identical to the solution ξ of (11.5.11) with initial data ω^0 *unless the initial vorticity field is radially symmetric*. From (11.5.11) we see that the core spreading algorithm accurately approximates the diffusion of vorticity but that the vorticity is advected by an averaged velocity field rather than by the local velocity. We should emphasize that it follows from Greengard's proof that the core spreading algorithm *does* approximate the correct equations when the vorticity field is radially symmetric.

We now describe a deterministic vortex method in $I\!R^2$ due to Cottet and Mas-Gallic [1990] that *has* been shown to converge to solutions of the Navier-Stokes equations. Let ω^0 denote an initial vorticity distribution and assume that the support of ω^0 is contained in some bounded set $\Omega_0 \subset I\!R^2$. Assume that a square grid of side h has been placed over Ω_0 and let

$$\tilde{\omega}^0(\mathbf{x}) = \sum_j \delta(\mathbf{x} - \tilde{\mathbf{x}}_j^0)\, \tilde{\omega}_j^0\, h^2\,,$$

be an approximation to ω^0 where, as usual, $\mathbf{x}_j^0$ is the center of the jth grid cell and $\tilde{\omega}_j^0 = \omega^0(\tilde{\mathbf{x}}_j^0)$.

In general, given the approximate particle positions $\tilde{\mathbf{x}}_j^k$ and their weights $\tilde{\omega}_j^k$ at time $t_k = k\Delta t$, the approximate vorticity field is given by,

$$\tilde{\omega}^k(\mathbf{x}) = \sum_j \delta(\mathbf{x} - \tilde{\mathbf{x}}_j^k)\, \tilde{\omega}_j^k\, h^2\,.$$

The approximation to the velocity field in the interval $t \in [t_k, t_{k+1}]$ is given by,

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_j (\mathbf{K} * G_{\Delta t})(\mathbf{x} - \tilde{\mathbf{x}}_j^k(t))\, \tilde{\omega}_j^k\, h^2\,,$$

where $G_{\Delta t}$ is the heat kernel (11.5.5) with $t = \Delta t$. To obtain the approximate particle paths $\tilde{\mathbf{x}}_i(t)$ for $t \in [t_k, t_k + \Delta t]$ we solve the following N ordinary differential equations,

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \tilde{\mathbf{u}}(\tilde{\mathbf{x}}_i(t), t)\,, \qquad t \in [t_k, t_{k+1}]\,, \tag{11.5.12a}$$

$$\tilde{\mathbf{x}}_i(t_k) = \tilde{\mathbf{x}}_i^k\,. \tag{11.5.12b}$$

This is simply a standard vortex method in the interval $[t_k, t_{k+1}]$ with $\mathbf{K}_\delta = \mathbf{K} * f_\delta$ replaced by $\mathbf{K} * G_{\Delta t}$. Now set $\tilde{\mathbf{x}}_j^{k+1} = \tilde{\mathbf{x}}_j(t_{k+1})$ to obtain an intermediate vorticity field of the form,

$$\tilde{\omega}^{k+\frac{1}{2}}(\mathbf{x}) = \sum_j \delta(\mathbf{x} - \tilde{\mathbf{x}}_j^{k+1})\, \tilde{\omega}_j^k\, h^2\,. \tag{11.5.13}$$

Finally, to update the vortex strengths $\tilde{\omega}_i^k \to \tilde{\omega}_i^{k+1}$ one explicitly solves the diffusion equation (11.5.3) with initial data (11.5.13) to obtain,

$$\tilde{\omega}_i^{k+1} = (G_{\Delta t} * \tilde{\omega}^{k+\frac{1}{2}})(\mathbf{x}_i^{k+1}) = \sum_j G_{\Delta t}(\mathbf{x}_i^{k+1} - \mathbf{x}_j^{k+1})\, \tilde{\omega}_j^k\, h^2 \,.$$

Note that in the description of the algorithm given here the particle paths $\tilde{\mathbf{x}}_i(t)$, being solutions of the ODEs (11.5.12a,b), are smooth functions of t whereas the vortex strengths $\tilde{\omega}_i^k$ are piecewise constant functions of t. Cottet and Mas-Gallic have proven that this version of the algorithm converges to solutions of the Navier-Stokes equations. Of course in actual practice one discretizes the system (11.5.12a,b) in time to obtain piecewise constant approximations to the $\tilde{\mathbf{x}}_i(t)$.

In the two dimensional version of the random vortex method the vortex strengths remain constant and the diffusion of vorticity is approximated by having the particle positions take a random walk with appropriate mean and variance. Here, on the other hand, the diffusion of vorticity is approximated by changing the particle strengths in a manner consistent with the exact solution of the heat equation. Also note that in Cottet and Mas-Gallic's method the approximate vorticity field is a collection of point vortices, but that the approximate velocity field is smoothed by convoluting it with the heat kernel $G_{\Delta t}$. This is not the same as the core spreading algorithm however. In the core spreading algorithm the smoothing function $f_t = G_t * f_0$ spreads out in time. In Cottet and Mas-Gallic's algorithm Δt plays the role of the cutoff parameter δ and is fixed for all time.

Cottet and Mas-Gallic [1990] have proven that this algorithm converges to solutions of the Navier-Stokes equations. We state their main result here. Let $\omega(\mathbf{x}, t)$ denote the exact solution of the Navier-Stokes equations (11.2.14a–d) in $\mathbb{R}^2$ with initial data ω^0 and boundary condition (11.2.2). Let $\mathbf{u}(\mathbf{x}, t) = (\mathbf{K} * \omega)(\mathbf{x}, t)$ and let $W^{m,p}(\mathbb{R}^2)$ denote the space of all functions $f : \mathbb{R}^2 \to \mathbb{R}$ such that $D^\alpha f \in L^p$ for all multi-indices α with $|\alpha| \le m$. Then we have the following,

Theorem 11.5.3 (Cottet and Mas-Gallic) *Assume that $\omega^0 \in W^{m,\infty}(\mathbb{R}^2) \cap W^{m,1}(\mathbb{R}^2)$ for any $m \ge 0$ and that h and Δt satisfy*

$$h \le C_0\, (\nu \Delta t)^{1/2+s} \tag{11.5.14}$$

where C_0 and s are arbitrary positive constants. Then given $T > 0$ and $\nu_0 > 0$ there exists a constant C_1 depending only on ω^0 and T such that for Δt small enough,

$$\max_{0 \le t \le T} \|\mathbf{u}(\cdot, t) - \tilde{\mathbf{u}}(\cdot, t)\|_p \le C_1\, \nu\, \Delta t$$

for all $2 < p < \infty$ and $\nu \le \nu_0$.

Note that since the solution of the heat equation $G_{\Delta t} * \tilde{\omega}^{k+\frac{1}{2}}$ has infinite support - even though $\tilde{\omega}^{k+\frac{1}{2}}$ does not – the total amount of vorticity in the flow is *not* conserved by this algorithm,

$$\int_{I\!R^2} \omega^0(\mathbf{x})\, d\mathbf{x} \approx \sum \tilde{\omega}_j^0 h^2 > \sum \tilde{\omega}_j^1 h^2 > \ldots > \sum \tilde{\omega}_j^k h^2 > \ldots .$$

(Nevertheless the method converges to solutions of the Navier-Stokes equations.) Choquin and Huberson [1989] have suggested the following modification which does conserve the total amount of vorticity in the flow,

$$\tilde{\omega}_i^{k+1} = \tilde{\omega}_i^k + \sum_j G_{\Delta t}(\tilde{\mathbf{x}}_i^{k+1} - \tilde{\mathbf{x}}_j^{k+1})\, (\tilde{\omega}_j^k - \tilde{\omega}_i^k) h^2$$

This is just one example of a class of deterministic vortex methods that have been considered by Cottet, Mas-Gallic, and their colleagues. We refer the interested reader to Choquin and Lucquin-Desreux [1988], Choquin and Huberson [1989], Cottet [1990 and 1991], Cottet and Mas-Gallic [1990], Lucquin-Desreux [1987], and Mas-Gallic [1990] for further details and references.

Fishelov [1990b] has studied a deterministic vortex method in which the diffusion of vorticity is also approximated by adjusting the particle weights rather than their positions. We briefly describe the three dimensional, time continuous version of this algorithm here. Suppose we are given an approximate vorticity distribution of the form

$$\tilde{\boldsymbol{\omega}}(\mathbf{x}, t) = \sum_j f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3 ,$$

where f_δ is some smoothing function. Then the associated velocity field is given by

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_j \mathbf{K}_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j^k)\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3 ,$$

with $\mathbf{K}_\delta = \mathbf{K} * f_\delta$. Consider the diffusive term $\nu \Delta \omega$ of equation (11.2.14a). Fishelov has made the observation that one can evaluate this term explicitly,

$$\nu \Delta \tilde{\boldsymbol{\omega}}(\mathbf{x}, t) = \nu \sum_j \Delta f_\delta(\mathbf{x} - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3 ,$$

provided that f_δ is chosen so that an explicit representation of Δf_δ can be found. This gives rise to the following coupled system of ordinary differential equations,

$$\frac{d\tilde{\mathbf{x}}_i}{dt}(t) = \sum_j \mathbf{K}_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \tilde{\boldsymbol{\omega}}_j(t)\, h^3 , \qquad (11.5.15a)$$

$$\tilde{\mathbf{x}}_i(0) = \mathbf{x}_i^0 , \qquad (11.5.15b)$$

and

$$\frac{d\tilde{\omega}_i}{dt}(t) = (\tilde{\omega}_i(t) \cdot \nabla_x) \sum_j \mathbf{K}_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \tilde{\omega}_j(t)\, h^3$$

$$+ \nu \sum_j \Delta f_\delta(\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_j(t))\, \tilde{\omega}_j(t)\, h^3\,, \qquad \text{(11.5.16a)}$$

$$\tilde{\omega}_i(0) = \boldsymbol{\omega}^0(\mathbf{x}_i^0)\,. \qquad \text{(11.5.16b)}$$

Equation (11.5.15a,b) is our usual vortex method approximation to

$$\boldsymbol{\omega}_t + (\mathbf{u} \cdot \nabla)\, \boldsymbol{\omega} = 0\,,$$

while equation (11.5.16a,b) is an approximation to

$$\boldsymbol{\omega}_t = (\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} + \nu \Delta \boldsymbol{\omega}\,.$$

One expects that, given the appropriate relationship between h and δ, the solution of equations (11.5.15a,b) and (11.5.16a,b) will converge to a function that satisfies the Navier-Stokes equations

$$\boldsymbol{\omega}_t + (\mathbf{u} \cdot \nabla)\, \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} + \nu \Delta \boldsymbol{\omega}\,, \qquad \text{(11.5.17)}$$

in the limit as $h, \delta \to 0$. Fishelov [1990b] has done some analysis of the accuracy and stability of this algorithm, but at the present time its convergence to solutions of (11.2.14a–d) or to the associated velocity field $\mathbf{u} = \mathbf{K} * \boldsymbol{\omega}$ remains open.

This is by no means an exhaustive list of the methods that have been developed as an alternative to the random walk. It is simply a representative sample chosen on the basis of the author's knowledge and experience. We refer the reader to the references listed above for further information on this topic.

11.5.5 A Few Remarks

We close this section with a few comments concerning what we believe are the important issues in the area of vortex methods for viscous flows. It is widely believed that in order to resolve the smallest relevant scales in a viscous flow a grid-based numerical method must satisfy $\Delta x = O(\sqrt{\nu})$ where Δx is the grid spacing in each space direction (e.g., see Henshaw, Kreiss, and Reyna [1991]). Thus, in order to maintain a given level of accuracy in n space dimensions, the computational cost will increase like $O(\nu^{-n/2})$ as $\nu \to 0$.

The random vortex method was specifically designed to circumvent this problem (e.g., see Chorin [1973]). The claim that is generally made with regards to this method – and vortex methods in general – is that because it is a particle method the "diffusive" errors found in grid-based methods are largely eliminated. Furthermore, since the particles naturally concentrate in regions of greatest vorticity, it is also claimed that vortex methods have a sort of a built in adaptivity – concentrating computational cost in regions of greatest interest. If these claims are true, then the computational cost required to maintain a fixed level of accuracy should not increase as $\nu \to 0$. However, these claims have never been rigorously substantiated. Now that the convergence of the random vortex method to solutions of the Navier-Stokes equations has been established it is time to carefully investigate the actual dependence of the error on the viscosity ν.

Similarly, it is often claimed that various deterministic vortex methods are superior to the random vortex method because they have a higher-order convergence rate. This is certainly the case for fixed ν. However, a very important issue that has yet to be addressed is "How does the error in a deterministic vortex method depend on ν?" Or equivalently, "How does the computational cost of a deterministic vortex method depend on ν?" For example, if the condition relating Δt, h, and ν in (11.5.14) cannot be eliminated, then – for ν small enough – it will be less expensive to use the random vortex method to compute a given flow to some fixed level of accuracy.

Finally, we remark that it may very well turn out to be the case that the answer to the above questions depends on the number of space dimensions n. For example, it is entirely possible that some restriction similar to (11.5.14) is necessary in three dimensions but not in two. It will be of great interest to have the answer to these questions in both two and three dimensions.

11.6 The No-Slip Boundary Condition

We now turn to a discussion of numerical techniques for satisfying the no-slip boundary condition (11.2.15) in a random vortex method computation. Chorin [1973] originally satisfied (11.2.15) by creating vortices at grid points on $\partial\Omega$ such that the velocity induced by these vortices cancelled the tangential component of the velocity along $\partial\Omega$. After further study Chorin [1978] proposed another particle creation algorithm known as the *vortex sheet method.* This method is based on approximating solutions of the Prandtl boundary layer equations in a thin region adjacent to $\partial\Omega$. The vortex sheet method has probably been the most widely used technique for satisfying the no-slip boundary condition in random vortex method computations (e.g., see Baden and Puckett [1990], Cheer [1983 and 1989], Choi, Humphrey, and Sherman [1988], Chorin [1980], Fishelov [1990a], Ghoniem, Chorin, and Oppenheim [1982], Sethian [1984], Sethian and Ghoniem

[1988], Summers, Hanson, and Wilson [1985], Tiemroth [1986], and Zhu [1989]). For this reason we will describe it in some detail below. Chorin's original vortex creation algorithm has not been as widely used. However, Anderson *et al.* [1990] have recently used a variant of Chorin's technique to study certain problems associated with vortex shedding past a cylinder. See also Leonard [1980] for a discussion of Chorin's original idea and references to other workers who have used it.

At the time of this writing no one has proved that the random vortex method together with some algorithm for satisfying the no-slip boundary condition converges to solutions of the Navier-Stokes equations. A complete analysis of the vortex sheet method is also an open problem. Some progress has been made in this area however. Marchiorio and Pulverenti [1983] constructed a sequence of stochastic processes which are similar to the vortex sheet method and showed that they converge to solutions of the Prandtl boundary layer equations. However their proof contains no convergence rates or error estimates that are of computational value. Hald [1986] has proved the convergence of a random walk algorithm with particle creation to solutions of a convection-diffusion equation. Puckett [1989] has proved a consistency result for the random walk and particle creation portion of the vortex sheet method. However, the stability and an analysis of the "inviscid" part of the algorithm remain open problems.

There has also been progress in related areas. For example, Mas-Gallic [1990] has recently proved the convergence of a two dimensional deterministic vortex method with boundary conditions. Also, as we mentioned earlier, Hou and Wetton [1990] have proved the convergence of a finite difference method for approximating solutions of the Navier-Stokes equations in vorticity-stream formulation with a vorticity boundary condition. We also refer the interested reader to Ghoniem and Sherman [1985] for an interesting discussion of random walk methods and particle creation algorithms. However, the issue of how best to satisfy the no-slip boundary condition in a vortex method computation is an important and largely unanswered question.

11.6.1 The Vortex Sheet Method

Let Ω denote a domain containing a viscous, incompressible fluid which has a solid boundary $\partial\Omega$ upon which the no-slip boundary condition (11.2.15) must be satisfied. In a hybrid random vortex / vortex sheet method computation the computational domain is divided into two regions: an interior (or exterior) region Ω_{NS} located away from $\partial\Omega$ and a *sheet layer* Ω_{Pr} located adjacent to $\partial\Omega$. In Ω_{NS} one uses the random vortex method to approximate solutions of the incompressible Navier-Stokes equations, while in Ω_{Pr} one uses the vortex sheet method to approximate solutions of the Prandtl boundary layer equations. We use the term sheet layer to distinguish the *computational* boundary layer from the *physical* boundary layer. While the justification for dividing the computational domain into the two regions Ω_{NS} and Ω_{Pr} is soundly based on the theory of boundary layers (e.g., see Schlichting [1979]), it is sometimes the case that the sheet method is used in regions where the underlying assumptions implicit in the use of the Prandtl equations are in doubt. However, even though many aspects of the combined random vortex / vortex sheet algorithm remain to be rigorously justified – such as the use of the vortex sheet method near points of separation – we emphasize that it has been successfully used by a number of workers to model a wide variety of flows.

Both the random vortex method and the vortex sheet method are particle methods. The particles carry concentrations of vorticity; the velocity field within each of the regions is uniquely determined by the particle positions and their concentrations together with the appropriate boundary conditions. Both methods are fractional step methods. One of the fractional steps consists of evolving the particles and their concentrations in this velocity field. The other step consists of letting the particle positions undergo a random walk to account for the diffusive effects of viscosity.

In Ω_{NS} the particles are called *vortices* or *vortex blobs*. In Ω_{Pr} they are called *vortex sheets*. The no-flow boundary condition is satisfied on $\partial\Omega$ by imposing a potential flow on Ω_{NS} which cancels the normal component of the velocity due to the vortices. The no-slip boundary condition is satisfied by creating vortex sheets on $\partial\Omega$ which subsequently participate in the flow. The two solutions are matched by converting sheets that leave the sheet layer into vortices with the same circulation, converting vortices that enter the sheet layer into sheets with the same circulation, and letting the velocity at infinity in the Prandtl equations be the tangential component of the velocity on $\partial\Omega$ due to the vortices in Ω_{NS}. The sheet creation process and subsequent movement of the sheets into the interior of the flow mimics the physical process of vorticity creation at a boundary and is one of the attractive features of this numerical method.

For simplicity we describe the vortex sheet method in $\mathbb{R}^2$. The generalization to $n = 3$ is straightforward (e.g., see Chorin [1980] or Fishelov [1990a]). Let (x, y) denote coordinates that are parallel and perpendicular to the boundary respectively. Let (u, v) denote the corresponding velocity components, ω the vorticity, and ν the kinematic viscosity. Assume that the boundary is located at $y = 0$ and let $U_\infty(x, t)$ denote the "velocity at infinity" which is imposed on the flow from outside the boundary layer. In vorticity formulation the Prandtl equations are

$$\omega_t + u\omega_x + v\omega_y = \nu\, \omega_{yy}\,, \tag{11.6.1a}$$

$$\omega = -u_y\,, \tag{11.6.1b}$$

$$u_x + v_y = 0\,, \tag{11.6.1c}$$

$$u(x, 0, t) = 0\,, \tag{11.6.1d}$$

$$v(x,0,t) = 0\,, \tag{11.6.1e}$$

$$\lim_{y\to\infty} u(x,y,t) = U_\infty(x,t)\,. \tag{11.6.1f}$$

Note that in the limiting process ($\nu \to 0$) by which one derives the Prandtl equations from the Navier-Stokes equations the vorticity $\omega = v_x - u_y$ has become $\omega = -u_y$ since $v_x = O(\sqrt{\nu}u_y)$.

In the vortex sheet method the vorticity at time $t = k\Delta t$ is approximated by a sum of linear concentrations of vorticity,

$$\tilde{\omega}^k(x,y) = \sum_j \omega_j b_l(x - x_j^k)\delta(y_j^k - y)\,. \tag{11.6.2}$$

Each term of the sum in (11.6.2) is referred to as a *vortex sheet.* The jth sheet has center (x_j^k, y_j^k) and *strength* or *weight* ω_j. Here δ is the Dirac delta function, and $b_l = b(x/l)$ is what we refer to as the *smoothing* or *cutoff function* in analogy with the vortex method. The most commonly used cutoff is the *hat* or *tent* function originally proposed by Chorin [1978],

$$b(x) = \begin{cases} 1 - |x| & |x| \le 1, \\ 0 & \text{otherwise.} \end{cases} \tag{11.6.3}$$

The parameter l is often referred to as the *sheet length* – even though the support of b_l is typically of length nl for some integer $n \ge 2$. Since b_l has finite support and since $\delta(y_j - y)$ is 0 for $y \ne y_j$, we see that the jth sheet is simply a line segment parallel to the boundary which carries a delta function concentration of vorticity. For b_l with b defined by (11.6.3) each sheet has length $2l$ and the vorticity concentration varies linearly along the length of the sheet – having a value of ω_j at the center and 0 at the ends. We briefly discuss other possible choices for b_l at the end of this section.

We can use (11.6.1b) and (11.6.1f) to write the tangential velocity in terms of the vorticity,

$$u(x,y,t) = U_\infty(x,t) + \int_y^\infty \omega(x,s,t)ds\,. \tag{11.6.4}$$

Our approximation to u at time $k\Delta t$ is determined by (11.6.2) and (11.6.4),

$$\tilde{u}^k(x,y) = U_\infty(x,k\Delta t) + \sum_j \omega_j b_l(x - x_j^k)\mathrm{H}(y_j^k - y)\,, \tag{11.6.5}$$

where $\mathrm{H}(y)$ is the Heaviside function

$$\mathrm{H}(y) = \begin{cases} 1 & y \ge 0, \\ 0 & \text{otherwise.} \end{cases}$$

From (11.6.5) we see that as one crosses the jth sheet in the vertical direction there is a jump in the tangential velocity $\tilde{u}$ of size $\omega_j b_l(x - x_j)$. This is the motivation for referring to the computational elements as vortex sheets. To find the velocity component normal to the boundary we first use (11.6.1c) and (11.6.1e) to write

$$v(x,y,t) = -\int_0^y u_x(x,s,t)ds\,. \qquad (11.6.6)$$

Then, by approximating u_x with a centered divided difference, we obtain our approximation to v,

$$\tilde{v}^k(x,y) = -\partial_x U_\infty(x,t)y - \frac{1}{l}\sum_j \omega_j(b_l(x^+ - x_j^k) - b_l(x^- - x_j^k))\min(y, y_j^k)\,. \quad (11.6.7)$$

where $x^+ = x + l/2$ and $x^- = x - l/2$.

Since $\tilde{u}^k$ and $\tilde{v}^k$ were constructed using (11.6.4) and (11.6.6) respectively the velocity field $(\tilde{u}^k, \tilde{v}^k)$ automatically satisfies equations (11.6.1b,c) and the boundary conditions (11.6.1e,f). Furthermore, given U_∞, this velocity field is completely determined by the sheet positions (x_j^k, y_j^k) and their strengths ω_j.

The vortex sheet method is a fractional step method. The first step is the numerical solution of the convective part of equation (11.6.1a)

$$\omega_t + u\omega_x + v\omega_y = 0\,. \qquad (11.6.8)$$

The second step is the numerical solution of the diffusive part of (11.6.1a)

$$\omega_t = \nu\omega_{yy} \qquad (11.6.9)$$

subject to the no-slip boundary condition (11.6.1d).

Given an approximation $(\tilde{u}^k, \tilde{v}^k)$ to the velocity field at the kth time step the velocity at the next time step is determined as follows. We first evaluate $(\tilde{u}^k, \tilde{v}^k)$ at the center of each sheet. Denote the velocity at the center of the jth sheet by $(\tilde{u}_j^k, \tilde{v}_j^k)$. Our numerical approximation to (11.6.8) is found by moving the center of each sheet one time step of length Δt in this direction to obtain

$$(x_j^{k+\frac{1}{2}}, y_j^{k+\frac{1}{2}}) = (x_j^k, y_j^k) + \Delta t(\tilde{u}_j^k, \tilde{v}_j^k)\,. \qquad (11.6.10)$$

In general, the sheet positions given by (11.6.10) induce a non-zero tangential velocity on the boundary which we denote by $\tilde{u}^{k+\frac{1}{2}}(x,0)$. In order to approximately satisfy the no-slip boundary condition (11.6.1d) we create sheets on the boundary. Let a_r, $r = 1, \ldots, M$

denote equally spaced grid points at $y = 0$ with grid spacing l: $a_{r+1} - a_r = l$, let $u_r = \tilde{u}^{k+\frac{1}{2}}(a_r, 0)$, and let ω_{max} denote a computational parameter called the *maximum sheet strength*. Then for each r we create $q_r = [|u_r|/\omega_{max}]$ sheets with centers $(a_r, 0)$ and strengths $-\text{sign}(u_r)\ \omega_{max}$. Here $[x]$ denotes the greatest integer less than or equal to x.

The numerical solution of the diffusion equation (11.6.9) is found by letting all sheets – new and old – undergo a random walk in the y direction, reflecting those that go below the boundary. Therefore, the new sheet positions at time $(k+1)\Delta t$ are given by

$$(x_j^{k+1}, y_j^{k+1}) = (x_j^{k+\frac{1}{2}}, |y_j^{k+\frac{1}{2}} + \eta_j|)$$

where the η_j are independent, Gaussian distributed random numbers with mean 0 and variance $2\nu\Delta t$.

We wish to make several comments regarding the sheet creation algorithm here. First, note that in our presentation of the algorithm all sheets have magnitude ω_{max} and that we create no sheets at a_r when $|u_r| < \omega_{max}$. Hence, the no-slip boundary condition is satisfied at a_r only up to order ω_{max}. Originally Chorin [1978 and 1980] created sheets at the rth grid point whenever $|u_r| \geq \epsilon$ for some $\epsilon \ll \omega_{max}$ such that $\omega_j \leq \omega_{max}$ for all j and the sum of the strengths of these sheets exactly cancel u_r. For example, ϵ might be chosen to be on the order of the computer's round off error. However, this algorithm creates more sheets than the one described above, and since the work required to compute $(\tilde{u}_j^k, \tilde{v}_j^k)$ at the center of each sheet is $O(lN^2)$ where N is the number of sheets in the flow, this greatly increases the computational cost of the algorithm. Furthermore, numerical experiments to compare the two sheet creation algorithms conducted by Puckett [1989a] and Zhu [1989] have shown that there is no tangible increase in the accuracy of the numerical approximation when this latter, more expensive, sheet creation algorithm is used.

The second point we would like to make concerns the manner in which the no-slip boundary condition is satisfied and its relation to the cutoff function b_l. As noted above, $\tilde{u}^{k+\frac{1}{2}}(x, 0)$ is, in general, non-zero. Ideally one would like to add some function to $\tilde{u}^{k+\frac{1}{2}}$ which can be represented by the sum of sheets, and which cancels $\tilde{u}^{k+\frac{1}{2}}(x, 0)$ at all points x on the boundary but leaves $\tilde{u}^{k+\frac{1}{2}}(x, y)$ unchanged for $y > 0$. In other words, we wish to find some function of the form $\sum \omega_j b_l(x - x_j)\text{H}(y_j - y)$ such that

$$\tilde{u}^{k+\frac{1}{2}}(x, y) + \sum_j \omega_j\ b_l(x - x_j)\ \text{H}(y_j - y) = \begin{cases} \tilde{u}^{k+\frac{1}{2}}(x, y) & y > 0, \\ 0 & y = 0. \end{cases}$$

In general this is not possible. However, one can find ω_j and (x_j, y_j) so that this holds exactly for $y > 0$ and within $O(l)$ for $y = 0$. For example, when b is defined by (11.6.3) choosing (x_j, y_j) to be the grid points $(a_r, 0)$ reduces the problem to that of finding the coefficients of a piecewise linear interpolant to $-\tilde{u}^{k+\frac{1}{2}}(x, 0)$ with node points at the a_r (e.g., see Schultz [1973]). In other words, we wish to find coefficients c_r such that

$$\sum_r c_r b_l(x - a_r) \approx -\tilde{u}^{k+\frac{1}{2}}(x, 0)\,. \tag{11.6.11}$$

For the piecewise linear basis functions b_l (i.e., with b given by (11.6.3)) it turns out that $c_r = -u_r$ is the correct choice, since then the left hand side of (11.6.11) is the usual piecewise linear interpolant of $-\tilde{u}^{k+\frac{1}{2}}$ at $y \equiv 0$. In actual practice we approximate the left hand side of this expression by creating q_r sheets at each point $x = a_r$ with strengths $\omega_r = \pm\omega_{max}$ such that $q_r\omega_r \approx -u_r$.

This idea can be generalized to make use of higher order interpolation procedures. For example, one can replace b_l with a basis function for cubic splines (e.g., see Schultz [1973] or DeBoor [1978]). In Puckett [1987] the author studied the effect that this type of smoothing function has on the accuracy and rate of convergence of the vortex sheet method. The interested reader is referred there for further details.

Several studies have been made of the accuracy with which the vortex sheet method approximates solutions of the Prandtl equations. The method was used to approximate Blasius flow in both Chorin [1978] and Puckett [1989a] and to approximate Falkner-Skan flow in Summers [1989]. In particular, Puckett [1989a] contains an extensive tabulation of the error in approximating Blasius flow as a function of the computational parameters Δt, l, and ω_{max} while Summers [1989] examines the computed solutions for a family of flows, some of which contain stagnation points or separation points.

11.6.2 Choosing the Computational Parameters

There are three computational parameters in the vortex sheet method: the time step Δt, the sheet length l, and the maximum sheet strength ω_{max}. The only generally agreed upon constraint that these parameters must satisfy is the so called "CFL" condition,

$$\Delta t\, U_{\max} \leq l \tag{11.6.12}$$

where $U_{\max} = \max U_\infty(x)$. The justification usually given for (11.6.12) is that one wants to ensure that sheets move downstream at a rate of no more that one grid point per time step. This is an *accuracy condition* (as opposed to a stability condition) which ensures that information propagating in the streamwise direction will influence all features in the flow which are at least $O(l)$.

We also propose another accuracy condition,

$$\Delta t\, \omega_{max} \leq Cl^2 \tag{11.6.13}$$

where C is a constant with dimensions $1/L$ and L is a typical length scale. (For example, L might be the length of the boundary.) This condition is a consequence of requiring that the degree with which we refine u as a function of y be of the same order as the degree with which we refine features in the streamwise direction, $O(U_{\max}/\omega_{max}) = O(L/l)$, and then using (11.6.12). Note that since

$$\frac{d}{dx}\omega_j b_l(x - x_j) = O(\omega_{max}/l)\,,$$

sheets induce local (non-physical) streamwise gradients in $\tilde{u}$ which are $O(\omega_{max}/l)$. Condition (11.6.13) relates the size of these gradients to the ratio $l/\Delta t$.

We wish the circulation Γ about a vortex element to remain the same when a sheet leaves the sheet layer and becomes a vortex or vice-versa. If b_l is the piecewise linear smoothing function with b given by (11.6.3), then this implies that

$$|\Gamma_j| = l\,\omega_{max}\,, \tag{11.6.14}$$

where Γ_j is given by (11.3.12). Equation (11.6.14) serves to relate the computational parameters h and ω_j discussed in Sections 11.3 and 11.5 to the sheet length l and maximum sheet strength ω_{max}.

In a hybrid random vortex method / vortex sheet method computation there are two computational parameters which still must be chosen: the cutoff radius δ and the sheet layer thickness ϵ. We would like to relate the cutoff radius δ to the vortex sheet parameters ω_{max} and l. Let us assume that we are using Chorin's cutoff function (11.3.10). Then the velocity kernel is given by (11.3.11). We seek δ so that a vortex at the edge of the boundary layer and its image (with respect to the boundary) with opposite sign will induce the same tangential velocity on the boundary as a sheet with the same position and strength. (Note that the sheet does not require an image since, by (11.6.6) and (11.6.7), the sheets satisfy the no-flow boundary condition (11.6.1e) exactly at $y = 0$.) If we set

$$\delta = \frac{l}{\pi}\,,$$

then we find that for $|x| < \delta$ a vortex at (x, y), together with its image at $(x, -y)$ will induce the same tangential velocity on the boundary at $(x, 0)$ as a sheet with center (x, y).

Recall that the random walks have standard deviation $\sqrt{2\nu\Delta t}$. One wishes to avoid having random walks which travel the length of the sheet layer in one time step and this principle is generally taken into account when choosing the sheet layer thickness ϵ. Usually ϵ is taken to be $\epsilon = C\sqrt{\nu\Delta t}$ for some constant C. Typically $C = 2$ or 3. This yields a boundary layer which has the appropriate scale, $O(\sqrt{\nu})$ (see Schlichting

[1979]). We reiterate however that ϵ represents the thickness of a numerical boundary layer which should be distinguished from the physical boundary layer.

11.7 Fast Vortex Methods

The velocity at the point $\mathbf{x}$ due to N vortices with positions $\mathbf{x}_j$ and weights Γ_j is given by

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \sum_{j=1}^{N} \mathbf{K}_\delta(\mathbf{x} - \mathbf{x}_j)\Gamma_j \, . \qquad (11.7.1)$$

The cost of evaluating this function at a single point $\mathbf{x}$ is $O(N)$ operations. In a vortex method computation one needs to evaluate (11.7.1) at each of the N positions $\mathbf{x}_j$ at least once per time step. (Several evaluations per time step are required for higher order time discretizations.) Therefore, the cost of computing one time step of the vortex method is $O(N^2)$ and this cost can become prohibitive as $N \rightarrow \infty$.

Similar considerations pertain to astronomical calculations of the force of gravity due to a large number of stars acting on one another or in computational models of plasmas in which electrons and ions interact. The underlying theme in each of these fields is that there are N objects, a potential function associated with each object, and one wishes to calculate the force on each object induced by the other $N - 1$ objects. The problem of evaluating the force due to N objects at each of the N objects is generically referred to as the "N-body problem". The direct evaluation of a sum of the type shown in (11.7.1) at each of the positions $\mathbf{x}_j$ is often referred to as the *direct method*. Techniques for reducing the cost of an N-body computation that arise in one field will often have applications to other fields. Thus there has been much cross-fertilization in this area of vortex methods. We refer the interested reader to the review article of L. Greengard [1990] for a discussion of the underlying similarities between these problems and for further references.

There are a variety of approximation techniques available for reducing the cost of computing (11.7.1) to $O(N \log N)$ or even $O(N)$. Because of their importance to the practical implementation of vortex methods – especially in three dimensions – the study of such fast vortex methods is currently an active area of research. We will briefly outline several ideas that have been introduced in the last decade for speeding up vortex method computations in two dimensions and indicate the direction that some of this research is taking in three dimensions.

For the purposes of illustrating some of the ideas that can be used to reduce the cost of the direct method we will assume that $\Omega = I\!R^2$. The stream function ψ associated with a *point* vortex at $\mathbf{x}_j$ with strength $\Gamma_j = \omega_j h^2$ is the solution of Poisson's equation,

$$\Delta\psi = -\omega \tag{11.7.2}$$

where

$$\omega(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_j)\Gamma_j\,, \tag{11.7.3}$$

and δ is the Dirac delta function. Hence, ψ is the fundamental solution of Poisson's equation in $I\!R^2$, centered at the point $\mathbf{x}_j$, and multiplied by the vortex strength Γ_j,

$$\psi(\mathbf{x}) = -\frac{1}{2\pi}\log(|\mathbf{x} - \mathbf{x}_j|)\,\Gamma_j\,. \tag{11.7.4}$$

The velocity $\tilde{\mathbf{u}}_j$ due to the vortex at $\mathbf{x}_j$ with strength Γ_j may be found by differentiating ψ,

$$\tilde{\mathbf{u}}_j(\mathbf{x}) = \nabla^{\perp}\psi(\mathbf{x}) \equiv (\psi_y(\mathbf{x}), -\psi_x(\mathbf{x}))\,. \tag{11.7.5}$$

Thus, if the velocity is thought of as a force field, then the stream function ψ plays the role of its potential. Analogously, in electrostatics the potential associated with a point charge at $\mathbf{x}_j$ with charge Γ_j is also given by (11.7.4) but now the electrostatic field due to this charge is given by

$$\mathbf{E}(\mathbf{x}) = -\nabla\psi(\mathbf{x}) = -(\psi_x(\mathbf{x}), \psi_y(\mathbf{x}))$$

One can readily see that computing the velocity due to a point vortex or the electrostatic field due to a point charge are equivalent problems. Similar analogies hold for the gravitational field due to a body at $\mathbf{x}_j$.

11.7.1 The Vortex-in-Cell Method

In essence the velocity field due to N vortices can be found by solving a linear elliptic problem. This observation can be exploited to reduce the cost of the computation by employing fast algorithms for the solution of elliptic problems. For example, one can place a grid over the support of the vorticity Ω_0, interpolate the values of the vorticity ω onto the grid,[5] solve (11.7.2) on the grid for ψ, and use divided differences to obtain an approximation to $\tilde{\mathbf{u}} = (\psi_y, -\psi_x)$.

This procedure yields values of the velocity field on the grid. To find the value of the velocity at a vortex center $\mathbf{x}_j$ we can interpolate the values of the velocity from the grid onto $\mathbf{x}_j$. Thus, if we have $N_g = N_x \times N_y$ grid points and an $O(N_g \log N_g)$ method for solving (11.7.2) on the grid, then the cost of evaluating the velocity at each

[5] When dealing with point vortices or "point charges" one can use a weighted area rule for the interpolation. (See e.g., Hockney and Eastwood [1981] or Leonard [1980].)

of the N vortices is now $O(N + N_g \log N_g)$. This idea was probably first introduced by Birdsall and Fuss [1969] within the context of computing plasma simulations. It is the simplest example of a "fast vortex method" and is commonly known as the *vortex-in-cell* method, or sometimes the *cloud-in-cell* method. See Baker [1979] and Christiansen [1973] for numerical applications of this technique to vortex method computations or see Leonard [1980] for a more detailed review of these ideas. Cottet [1987] has proven the convergence of a vortex-in-cell method applied to a vortex blob method.

The problem with the vortex-in-cell method is a loss of accuracy due to the approximate solution of (11.7.2) on the grid, the two interpolation steps, and the divided difference approximation to $\tilde{\mathbf{u}}$. For example, suppose that we use the vortex-in-cell method to approximate the velocity field due to a single point vortex at $\mathbf{x}_j$; i.e., to solve (11.7.2) with the right hand side given by (11.7.3). A close examination of the error in this procedure will show that the error in approximating $\tilde{\mathbf{u}}_j$ at some point $\mathbf{x}$ increases as the distance between $\mathbf{x}$ and $\mathbf{x}_j$ decreases. This is because the constants in the error bounds depend on ψ and its derivatives and these become unbounded as $\mathbf{x} \to \mathbf{x}_j$. Thus, the order of the error will remain the same for all $\mathbf{x}$, but the actual computed error will increase substantially as $\mathbf{x}$ approaches $\mathbf{x}_j$.

This loss of accuracy as $\mathbf{x} \to \mathbf{x}_j$ can be ameliorated by only considering the influence of "far away" vortices when approximating the velocity field at a point $\mathbf{x}$ from values on the grid. For vortices near $\mathbf{x}$ the velocity at $\mathbf{x}$ due to those vortices is computed exactly

$$\tilde{\mathbf{u}}^{nearby}(\mathbf{x}, t) = \sum_{|\mathbf{x}-\mathbf{x}_j|<C} \mathbf{K}_\delta(\mathbf{x} - \mathbf{x}_j)\Gamma_j \,. \tag{11.7.6}$$

Note that this sum is only over those vortices "near" $\mathbf{x}$. Methods of this type have been used for plasma simulations by Hockney, Goel and Eastwood [1974] as well as others. They are commonly called particle-particle/particle-mesh (P^3M) methods. See Hockney and Eastwood [1981] for a review of this and similar methods.

In some sense all "accurate" fast vortex methods have a radius C of the form that appears in equation (11.7.6). Outside this radius the velocity field due to a vortex may be accurately calculated using some fast approximation technique; inside this radius the velocity field must be computed directly.

11.7.2 The Method of Local Corrections

An interesting variant of the particle-particle/particle-mesh idea, known as the method of local corrections, was proposed by Anderson [1986]. The method of local corrections was designed so that it could be used with high order accurate velocity kernels such as those described in Section 11.3.4. It is based on the observation that the velocity field due to a point vortex is harmonic away from the center of the vortex,

$$\Delta \tilde{\mathbf{u}}_j = 0 \qquad \text{for} \qquad \mathbf{x} \neq \mathbf{x}_j\,,$$

where $\tilde{\mathbf{u}}_j = \mathbf{K}(\mathbf{x} - \mathbf{x}_j)\Gamma_j$ is the velocity field due to a point vortex centered at $\mathbf{x}_j$ (c.f. equation (11.7.5)).

Suppose that we are given a grid with mesh spacing h_1 and an rth order accurate approximation to the Laplacian on this grid denoted by Δ^{h_1}. (It is not necessary that this grid be identical to the grid – with spacing denoted by h - that is used to determine the initial data. In fact, optimum accuracy and efficiency may very well be achieved when $h_1 \neq h$.) Rather than solving (11.7.2) for the stream function ψ, Anderson solves a Poisson equation for the velocity field $\tilde{\mathbf{u}}$,

$$\Delta \tilde{\mathbf{u}}^{h_1} = \sum_j \mathbf{g}_{D_j} \tag{11.7.7}$$

where the right hand side is the sum of approximations to $\Delta \tilde{\mathbf{u}}_j$ on the grid

$$\mathbf{g}_{D_j}(\mathbf{x}) = \begin{cases} \Delta^{h_1} \tilde{\mathbf{u}}_j & |\mathbf{x} - \mathbf{x}_j| < D, \\ 0 & |\mathbf{x} - \mathbf{x}_j| \geq D. \end{cases} \tag{11.7.8}$$

Since $\Delta \tilde{\mathbf{u}} = 0$ for $\mathbf{x} \neq \mathbf{x}_j$, we can choose D so that $\mathbf{g}_{D_j}$ is an rth order accurate approximation to $\Delta \tilde{\mathbf{u}}_j$ on the grid. Anderson calls the constant D the *spreading radius*. The point of using the approximation $\mathbf{g}_{D_j}$ to $\Delta \tilde{\mathbf{u}}_j$ is that for small D the right hand side of (11.7.7) can be computed in only $O(N)$ operations rather than in the $O(NN_g)$ operations required to approximate $\Delta \tilde{\mathbf{u}}_j$ at each of the N_g grid points.

Once the right hand side of (11.7.7) is available we use our favorite fast Poisson solver to approximate the solution of (11.7.7) on the grid. This results in an approximation $\tilde{\mathbf{u}}^{h_1}$ on the grid to the velocity field $\tilde{\mathbf{u}}$ given by (11.7.1). In particular, note that by solving for $\tilde{\mathbf{u}}^{h_1}$ directly we have eliminated the error due to approximating $\tilde{\mathbf{u}} = (\psi_y, -\psi_x)$ with divided differences of ψ on the grid. Instead we now must compute the $\mathbf{g}_{D_j}$. Numerical experiments by Anderson [1985], Baden [1987], and Buttke [1991] have demonstrated that for $D = h_1$ or $D = 2h_1$ the approximation $\tilde{\mathbf{u}}^{h_1}$ to $\tilde{\mathbf{u}}$ on the grid is highly accurate.

The approximation $\tilde{\mathbf{u}}^{h_1}$ is next interpolated back onto the vortex positions. In the interpolation step Anderson cleverly exploited the fact that in two dimensions the two components of the velocity field $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})$ due to the sum of point vortices centered at points $\mathbf{x}_j$ are the real and complex parts of a complex analytic function,

$$F(\mathbf{x}) = \tilde{u}(z) - i\tilde{v}(z)\,. \tag{11.7.9}$$

Here we have identified $I\!R^2$ with the complex plane C: $\mathbf{x} = (x, y) \to z = x + iy$. Thus, one can use Lagrange interpolation in the complex plane to obtain a fourth order interpolation at only four points. In contrast, a bilinear interpolation at four points for each of the velocity components separately would only be second order. Lagrange interpolation at more points will result in higher order interpolation formulas for the velocity. However it should be noted that because this idea is based on a representation of the velocity as a function in the complex plane it does not generalize easily to three dimensions. In fact, the task of finding high order interpolation formulas is an important issue when generalizing the method of local corrections to three dimensions (e.g., see Buttke [1991]).

The last, and very essential part of this procedure is to *correct* the velocity field $\tilde{\mathbf{u}}^{h_1}$ before interpolating it back onto the individual vortices. This is accomplished as follows. Suppose we wish to calculate the velocity at $\mathbf{x}$ by interpolating $\tilde{\mathbf{u}}^{h_1}$ from the points $\boldsymbol{\alpha}_i$, $i = 1, \ldots, m$ on the grid. Let the vortices which satisfy $|\mathbf{x} - \mathbf{x}_j| < C$ be the "nearby" vortices. (Anderson calls the distance C the *correction radius*.) For each i we evaluate the velocity at $\boldsymbol{\alpha}_i$ induced by all nearby vortices and subtract this velocity from $\tilde{\mathbf{u}}^{h_1}(\boldsymbol{\alpha}_i)$. The resulting values of $\tilde{\mathbf{u}}^{h_1}$ at the $\boldsymbol{\alpha}_i$ are the ones used to interpolate the velocity onto the point $\mathbf{x}$. Finally, the exact velocities due to the vortices near $\mathbf{x}$ – i.e., the expression in (11.7.6) – are added to this value of the velocity at $\mathbf{x}$ to obtain our complete approximation to $\tilde{\mathbf{u}}$ at $\mathbf{x}$.

It is important to note that all of the steps in this procedure but the very last are performed as if the vortices at the points $\mathbf{x}_j$ are *point vortices*. Only when the velocity at $\mathbf{x}$ is corrected by (11.7.6) is the velocity field due to a vortex blob at the $\mathbf{x}_j$ used. If the smoothing function f_δ has compact support, say within a disk of radius R, then the velocity field due to a vortex blob at $\mathbf{x}_j$ and a point vortex at $\mathbf{x}_j$ are identical at distances greater than R from $\mathbf{x}_j$. Thus the method of local corrections will accurately represent the influence of the core function f_δ provided that $R \leq C$.

As with the vortex-in-cell method the cost of the method of local corrections is also $O(N + N_g \log N_g)$. It is somewhat more complex to program than a vanilla vortex-in-cell method, but considerably more accurate. Applications of this technique to both vector and parallel computers have been studied by Baden [1987] and a version for viscous flow with solid boundaries has been implemented by Baden and Puckett [1990]. We also note that Mayo [1985] employed similar ideas in the solution of Laplace's equation in irregular regions.

11.7.3 Tree Codes, Multipole Expansions, and the Fast Multipole Method

We now turn to a somewhat different set of techniques for reducing the cost of directly calculating (11.7.1) in two dimensions. These techniques rely on representing the velocity field due to a cluster of vortices as a truncated power series in z^{-1} which is valid sufficiently far from the cluster. (Once again we are identifying $I\!\!R^2$ with the complex plane C.) By combining these expansions with a hierarchical data structure which consists of nested boxes covering the computational domain, one can devise an algorithm of arbitrary accuracy which is $O(N \log N)$. Several clever modifications of this idea due to Greengard and Rokhlin [1987] can be made to reduce the cost of such a method to $O(N)$. We outline the methods as they would be applied to a point vortex computation and indicate at the end the modifications necessary for working with vortex blobs. The following discussion is loosely based on the review article of Greengard [1990]. For further details and a description of these methods in a more general setting see Greengard and Rokhlin [1987] or Greengard [1990].

The Tree Data Structure For the sake of simplicity let us assume that there are $N = 4^\gamma$ vortices uniformly distributed in a square box of side 1 which we will call Ω. We begin by describing the hierarchical data structure. Let us call the initial box level 0. Now subdivide this box into 4 equal boxes of side $1/2$ and call this level 1. In general, level l is obtained from level $l-1$ by subdividing each box at level l into 4 smaller boxes of equal size. Continue this procedure until at the lowest level of refinement we have N boxes. Thus, at level 0 there is 1 box, at level 1 there are 4 boxes and, in general, at the lth level there are 4^l boxes. Given a box A at level $l-1$, the boxes at level l which are obtained by subdividing box A are called the *children* of box A. A box is the called the *parent* of its children.

Definition 1 Two boxes are said to be *near neighbors* if they are at the same level of refinement and share a boundary point. (A box is a near neighbor of itself.)

Definition 2 Two boxes are said to be *well separated* if they are at the same level of refinement and not near neighbors.

Definition 3 With each box A is associated an *interaction list* which consists of all of the children of the near neighbors of A's parent that are well separated from A.

The interaction list is the key concept here. As we shall see, it enables one to organize the computation in an efficient manner. Note that at levels 0 and 1 all boxes are near neighbors of each other and hence each box's interaction list is empty. Note also that at any level of refinement the maximum size of the interaction list is 27. (It is smaller for boxes that lie on the boundary of the computational domain.)

The purpose of this tree of boxes is twofold. First, it provides a mechanism for determining which vortices are well separated from a given box A and thus amenable to a far field approximation of the velocity due to the vortices in box A. Second, it provides a mechanism, namely the interaction list, for keeping track of those vortices that have not yet had their interactions with the vortices in box A computed.

Multipole Expansions We now turn to the issue of fast far field approximations for computing the velocity between vortices in boxes that are well separated from one another. In order to describe this approximation it is again helpful to identify $\mathbb{R}^2$ with the complex plane C: $\mathbf{x} = (x, y) \to z = x + iy$ and view the velocity as the real and imaginary parts of the function F in the complex plane given by (11.7.9). If we substitute the expression for $\mathbf{K}$ given by (11.2.8) into (11.7.1) (with $\mathbf{K}_\delta$ replaced by $\mathbf{K}$ since we are considering point vortices) we find that the velocity field due to a cluster of m^6 point vortices is

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = -\frac{1}{2\pi}\sum_{j=1}^{m}\Gamma_j\frac{(y - y_j, x_j - x)}{(x - x_j)^2 + (y - y_j)^2} \quad .$$

Hence, F has the form

$$F(z) = \frac{1}{2\pi i}\sum_{j=1}^{m}\Gamma_j\frac{1}{(z - z_j)}$$

where we have identified $\mathbf{x}_j \in \mathbb{R}^2$ with $z_j \in C$. Note that F is an analytic function for $z \neq z_j$.

Now consider a cluster of vortices located at points z_j all of which are contained in a box A with center z_A such that $|z_j - z_A| < R$ for all j. Let z be such that $|z - z_A| > 2R$. Then $F(z)$ can be written as a *multipole expansion* about the point $z = z_A$,

$$F(z) = \sum_{k=0}^{\infty}\frac{a_k}{(z - z_A)^{k+1}} \qquad (11.7.10)$$

where the a_k are given by

$$a_k = \frac{1}{2\pi i}\sum_{j=1}^{m}\Gamma_j(z_j - z_A)^k \qquad k = 0, 1, 2, \ldots \quad . \qquad (11.7.11)$$

Note that the cost of forming each coefficient a_k is $3 \cdot m$ operations. (The $2\pi i$ can be divided into the strengths Γ_j initially to avoid needless divisions later.) Hence, the cost of forming the coefficients for the first p terms of (11.7.10) is $O(mp)$.

We can derive bounds for the error in using the first p terms of (11.7.10) as follows. Let

$$c = \frac{R}{|z - z_A|} \qquad \text{and} \qquad W = \frac{1}{2\pi}\sum_{j=1}^{m}|\Gamma_j| \, . \qquad (11.7.12)$$

[6] Note the use of the integer m rather than N. This is a notational convenience so that we may later consider the cost of taking many such expansions when the total number of vortices is $N > m$.

Then

$$
\begin{aligned}
\left|F(z) - \sum_{k=0}^{p} \frac{a_k}{(z-z_A)^{k+1}}\right| &= \left|\sum_{k=p+1}^{\infty} \frac{a_k}{(z-z_A)^{k+1}}\right| \\
&\leq W \sum_{k=p+1}^{\infty} \frac{R^k}{|z-z_A|^{k+1}} \\
&\leq \frac{W}{R} \sum_{k=p+1}^{\infty} \frac{R^{k+1}}{|z-z_A|^{k+1}} \\
&\leq \frac{W}{R} \left|\frac{R}{z-z_A}\right|^{p+2} \sum_{k=0}^{\infty} \frac{R^k}{|z-z_A|^k} \\
&\leq \frac{W}{R} \left|\frac{R}{z-z_A}\right|^{p+2} \frac{1}{1-c} \\
&\leq \frac{W}{R} \frac{1}{c^{-1}-1} c^{p+1} \\
&\leq \frac{W}{R} \left(\frac{1}{2}\right)^{p+1}
\end{aligned}
\tag{11.7.13}
$$

since, by assumption, $c < 1/2$.

In practice R is half the diagonal length of a box and hence $R \to 0$ as the level of refinement increases. However one can control the size of W/R as follows. In a typical (two dimensional) vortex method computation Ω is divided into $N = 4^\gamma$ squares each of side $h = 1/\sqrt{N} = 2^{-\gamma}$. Let z_j, $j = 1, \ldots, N$ denote the centers of the squares. Let $\omega(z)$ denote the vorticity at time $t = 0$ and let $\|\omega\|_\infty$ denote the sup norm of $\omega(z)$ in Ω. The initial data consists of N vortices, with positions z_j and strengths $\Gamma_j = \omega(z_j)\, h^2$.

At any given level l there are 4^l boxes and hence, there are $m_l = N/4^l = 4^{\gamma-l} = 2^{2(\gamma-l)}$ vortices per box. Furthermore, the side of a box at level l has length 2^{-l} and hence the distance from the center of a box to any corner is

$$
\begin{aligned}
\left|F(z) - \sum_{k=0}^{p} \frac{a_k}{(z-z_A)^{k+1}}\right| &\leq \frac{W}{R_l} \left(\frac{1}{2}\right)^{p+1} \\
&= \frac{1}{2\pi} \frac{2}{\sqrt{2}} 2^l \sum_{j=1}^{m_l} |\Gamma_j| \left(\frac{1}{2}\right)^{p+1} \\
&\leq \frac{1}{\sqrt{2}\pi} 2^l\, m_l\, \|\omega\|_\infty\, h^2 \left(\frac{1}{2}\right)^{p+1} \\
&= \frac{1}{\sqrt{2}\pi} 2^{2\gamma-l}\, \|\omega\|_\infty\, \frac{1}{N} \left(\frac{1}{2}\right)^{p+1}
\end{aligned}
\tag{11.7.14}
$$

$$\leq \frac{1}{\sqrt{2\pi}} \|\omega\|_\infty \left(\frac{1}{2}\right)^{p+1}.$$

since $2^{2\gamma - l} \leq N$ for all l. This estimate holds at all levels of refinement l.

Tree Codes One can combine the tree data structure and the multipole expansions (11.7.10) to obtain an $O(N \log N)$ algorithm for approximating (11.7.1) at each of the N vortices as follows. For the time being let us assume that $p = 2$.

Starting at level 2 form the multipole expansion coefficients given by (11.7.11) for each box A and use them to evaluate the velocity at all vortices which are contained in A's interaction list.[7] (Since there are no well separated boxes at levels 0 or 1 there is no need to perform this step at these levels.) So far we have calculated the influence of the vortices in box A on all the other vortices except those contained in A's near neighbors. To compute these interactions we recursively repeat the above procedure at successive levels of refinement. At each level calculate the coefficients (11.7.11) for every box A and evaluate the resulting multipole expansion at each vortex in A's interaction list. At the last level we also evaluate the interactions between each box and its nearest neighbors *directly*.

The total amount of work in this procedure is $O(N \log N)$. To see this first note that since there are N vortices and each vortex contributes to $p = 2$ expansion coefficients, it takes $3 \cdot p \cdot N = 6 \cdot N$ operations to form the coefficients (11.7.11) at each level. From the point of view of a vortex in box A at level l the cost of evaluating its velocity due to the expansions in the boxes in its interaction list is at most $(3 \cdot p + 1) \cdot 27 = 7 \cdot 27$ operations since there are at most 27 members of the interaction list and it takes 7 operations to evaluate the expansion (11.7.10) at a point z when $p = 2$. (The term $(z - z_A)^{-1}$ need only be evaluated once for each z.) Since there are N vortices the cost of evaluating all of the expansions at a given level is therefore $7 \cdot 27 \cdot N$. Thus, the cost of this procedure at any given level is $7 \cdot 27 \cdot N + 6 \cdot N = 195 \cdot N$ operations. At the last level there are at most 8 nearest neighbors and 1 vortex per box (by assumption the vortices are uniformly distributed in the computational domain). Thus, the cost of evaluating the direct interactions at the last level is $8N$ operations. (Self interactions are not computed.) Finally, there are $\log_4 N$ levels so the total amount of work is $O(N + N \log N) = O(N \log N)$ as claimed.

Of course, in general the vortices are not uniformly distributed in the computational domain. In practice one can implement an adaptive version of this algorithm that at each level checks every box to see if it contains any vortices. If not, this box is pruned

[7] For convenience we will often refer to the vortices contained in the boxes of A's interaction list as being in A's interaction list.

from the tree structure and ignored at subsequent levels. For most practical problems the running time of this algorithm will remain $O(N \log N)$. However the constant of proportionality will depend on the distribution of vortices in Ω. See Carrier, Greengard and Rokhlin [1988] for an example of such an algorithm.

Algorithms of this type have been used extensively in astrophysical calculations and are commonly called *tree codes*. See Apple [1985] or Barnes and Hut [1986] for a more complete discussion of tree codes. By taking $p > 2$ we will have a more accurate algorithm. This is essentially the approach taken by van Dommelen and Rundensteiner [1989]. Their algorithm is still $O(N \log N)$ but now the constant of proportionality depends on the precision desired; i.e., on p.

The Fast Multipole Method Greengard and Rokhlin [1987] have made several observations that allows one to construct an algorithm with $O(N)$ running time. These ideas are similar to those used by Rokhlin [1985] for the rapid solution of integral equations. We will briefly describe the Greengard-Rokhlin algorithm here. The interested reader is referred to Greengard and Rokhlin [1987] for further details and for proofs of the statements made below.

To begin, suppose we are given the multipole expansion (11.7.10) for $F(z)$ about the center z_A of some box A. Then we can expand F in a multipole expansion about the center z_P of A's parent P,

$$F(z) = \sum_{l=0}^{\infty} \frac{b_l}{(z - z_P)^{k+1}} .$$

This expansion will be valid to the accuracy stated in (11.7.13) provided that it is only evaluated at points z in boxes that are well separated from P. The coefficients b_l may be formed from the (already known) a_k,

$$b_l = \sum_{k=0}^{l} \binom{l}{k} a_k \, (z_A - z_P)^{l-k} \tag{11.7.15}$$

where $\binom{l}{k}$ are the binomial coefficients. Thus, if the coefficients a_k, $k = 1, \ldots, p$ of the first p terms of each child's multipole expansions are known, then one can obtain the coefficients b_l, $l = 1, \ldots, p$ of the parent's multipole expansion in $O(p^2)$ operations.

Next note that the multipole expansion (11.7.10) of $F(z)$ is analytic outside the circle of radius R centered at z_A. Therefore, $F(z)$ can be expanded in a Taylor series about any point z_B that satisfies $|z_B - z_A| > R$. In particular, suppose z_A and z_B are the centers of boxes A and B and that A and B are well separated from one another; i.e.,

$|z_B - z_A| > 2R$, where R is the half diagonal of a box. Then $F(z)$ can be expanded in a Taylor series about z_B that is valid in a circle of radius R centered at z_B,

$$F(z) = \sum_{l=0}^{\infty} c_l \, (z - z_B)^l \, .$$

Greengard and Rokhlin call this the *local expansion* in B of the velocity field due to the vortices at the $z_j \in A$. The important thing to notice here is that the coefficients c_l may also be written in terms of the coefficients a_k,

$$c_l = \frac{1}{(z_A - z_B)^{l+1}} \sum_{k=0}^{\infty} \binom{l+k}{k} \frac{a_k}{(z_A - z_B)^k} \, (-1)^{k+1}$$

In practice, to obtain pth order accuracy one truncates the series on the right hand side of this expression after p terms. Thus, given the multipole expansion coefficients a_k for $k = 1, \dots, p$, we can compute pth order approximations $\tilde{c}_l$ to the local expansion coefficients c_l, $l = 1, \dots, p$ of the form

$$\tilde{c}_l = \frac{1}{(z_A - z_B)^{l+1}} \sum_{k=0}^{p} \binom{l+k}{k} \frac{a_k}{(z_A - z_B)^k} \, (-1)^{k+1} \tag{11.7.16}$$

in $O(p^2)$ operations.

Finally, we have a formula similar to (11.7.15) for shifting the local expansion about the center of a given box A onto the centers of A's children. Let z_A denote the center of box A at level $l - 1$ and let z_C denote the center of one of A's children. Then

$$\sum_{l=0}^{p} c_l (z - z_A)^l = \sum_{l=0}^{p} \left(\sum_{k=l}^{p} c_l \binom{k}{l} (z_C - z_A)^{k-l} \right) (z - z_C)^l \, . \tag{11.7.17}$$

Note that this formula is exact. The left hand side is simply a polynomial in powers of $(z - z_A)$ and the right hand side is simply the same polynomial in powers of $(z - z_C)$.

We are now in position to describe the fast multipole method. The method consists of two passes, beginning with an upward pass from the finest level of boxes up to level 2. This is the coarsest level that still has boxes with nonempty interaction lists. We then retrace our steps going back down through the tree hierarchy, until at the finest level of refinement each box has a local expansion that represents the velocity field due to all of the vortices except near neighbors.

Upward Pass: Starting at the finest level we form the coefficients (11.7.11) for every box. For each parent P at level l we take the coefficients formed at level $l+1$ for each of

P's children and using (11.7.15) shift them on to P's center. By adding the contributions from each of P's children we form the coefficients of P's multipole expansion. At each level we save the coefficients associated with each box for the later use.

Downward Pass: Given a box A at level 2 we use (11.7.16) to convert the coefficients of the multipole expansions associated with each box in A's interaction list to a local expansion about A's center and add these coefficients together. Before moving to the next level down we use (11.7.17) to shift these coefficients onto the center of each of A's children. We then recursively repeat the above procedure: In each box A at level l we have the coefficients of a local expansion that represents the velocity field due to all of the vortices in boxes that are well separated from A's parent. To these coefficients we add the coefficients for the local expansions in A's interaction list. At the end of this procedure each box at the finest level will contain the coefficients of a local expansion that represents the velocity field due to all of the vortices except its nearest neighbors – their contribution to the velocity field will be evaluated directly. Thus, to obtain the velocity at each vortex we evaluate the velocity field given by the appropriate local expansion and add to it the exact velocity due to its nearest neighbors.

The total cost of the above procedure is $O(Np^2)$. To see this first note that the cost of forming all of the local expansion coefficients at the finest level is $O(Np)$. The cost of shifting the multipole coefficients from a child onto its parent is $O(p^2)$ and since this must be done at most N times (there are at most N children) the cost of the remainder of the upward pass is $O(Np^2)$. Similarly, one can show that the cost of the downward pass is also $O(Np^2)$. At the finest level the cost of evaluating the appropriate local expansion at each vortex is $O(Np)$ and the cost of computing the nearest neighbor interactions directly is $O(N)$.

Some care should be taken in the programming of this algorithm to achieve the maximum possible savings. We refer the interested reader to Greengard and Rokhlin [1988] for details. Greengard and Rokhlin [1987] have also studied the implementation of various boundary conditions in conjunction with the fast multipole method. We also remark that Greengard [1988] has devised a fast technique along similar lines for computing the flow due to vortices in a channel.

Just as with the method of local corrections we can use this algorithm to evaluate the velocity field due to a collection of vortex blobs. For example, if the core functions have support contained in a region of radius R, then we must choose the maximum level of boxes so that at all levels $|\mathbf{x}_i - \mathbf{x}_j| < R$ implies that the vortices at $\mathbf{x}_i$ and $\mathbf{x}_j$ are nearest neighbors. It is conceivable that one might be able to find core functions with large or even infinite support, but which induce a velocity that is sufficiently close to that induced by a point vortex at distances greater than R from the vortex center. For such core functions both the method of local corrections and fast multipole expansions might then be applicable. However the author is not aware of any published work concerning this issue.

11.7.4 Fast Vortex Methods in Three Dimensions

The number of vortices that are required to retain a given level of accuracy, of say $O(h^r)$, increases from $O(h^{-2})$ in two dimensions to $O(h^{-3})$ in three. Therefore, fast algorithms are especially crucial in three dimensional vortex method computations. The generalization to three dimensions of the ideas illustrated above is currently an active area of research. In what follows we will simply mention recent work and give the appropriate references.

Buneman, Couet and Leonard [1981] have studied a three dimensional version of the vortex-in-cell method. A discussion of some of the salient ideas may be found in the review articles of Leonard [1980 and 1985]. Greengard and Rokhlin [1988a] have implemented a three dimensional version of their fast multipole method which is based on spherical harmonics. Buttke [1991] has also studied the fast multipole method in three dimensions. He has concluded that in three dimensions it is best to eliminate the hierarchy of expansions. This results in a method which is $O(N^{3/2})$ rather than $O(N \log N)$ or $O(N)$. Buttke presents evidence that for many three dimensional applications this technique will be less expensive than the full fast multipole method. However it should be noted that Greengard and Rokhlin [1988b] have devised strategies which will significantly reduce the cost of the 3D multipole algorithm.

Buttke [1991] also describes a three dimensional implementation of the method of local corrections and compares its performance to his version of the fast multipole method. Almgren, Buttke, and Colella [1991] have also implemented a three dimensional version of the method of local corrections. Their implementation includes an adaptive mesh refinement algorithm based on a multigrid method.

Finally we note that Anderson [1990] has designed a method which shares many of the features of the fast multipole method. However Anderson's method is based on Poisson's formula for representing solutions of Laplace's equation as a boundary integral over a disk or sphere rather than on multipole or spherical harmonic expansions. According to Anderson this technique has the advantage that the computer program for the method in two dimensions can easily be modified for three dimensions. He also discusses how to exploit the essential structure of a multigrid code to quickly develop a version of this method in either two or three dimensions.

References

[1] Alessandrini, G., Douglis, A., and Fabes, E. [1983]. An Approximate Layering Method for the Navier-Stokes Equations in Bounded Cylinders. *Ann. Mat. Pura Appl.* **135**, 329–347.

[2] Almgren, A., Buttke, T. and Colella, P. [1991]. A Fast Vortex Method in Three Dimensions. *Proceedings of the 10th AIAA Computational Fluid Dynamics Conference.* Honolulu, HI, 10 pages.

[3] Anderson, C. R. [1985]. A Method of Local Corrections for Computing the Velocity Field Due to a Distribution of Vortex Blobs. *J. Comp. Phys.* **61**, 111–123.

[4] Anderson, C. R. [1986]. A Vortex Method for Flows with Slight Density Variations. *J. Comp. Phys.* **62**, 417–432.

[5] Anderson, C. R. [1988]. Observations on Vorticity Creation Boundary Conditions. *Mathematical Aspects of Vortex Dynamics* (Ed. by R. E. Caflisch), S.I.A.M., Philadelphia, 144–159.

[6] Anderson, C. R. [1989]. Vorticity Boundary Conditions and Boundary Vorticity Generation for Two-Dimensional Viscous Incompressible Flows. *J. Comp. Phys.* **80**, 72–97.

[7] Anderson, C. R. [1990]. The Fast Multipole Method without Multipoles. *UCLA Computational and Applied Mathematics*, Report **90-14**, 1–28.

[8] Anderson, C. R. and Greengard, C. [1985]. On Vortex Methods. *SIAM J. Numer. Anal.* **22**, 413–440.

[9] Anderson, C. R. and Greengard, C. [1989]. The Vortex Ring Merger Problem at Infinite Reynolds Number. *Commun. Pure Appl. Math.* **42**, 1123–1139.

[10] Anderson, C. R., Greengard, C., Greengard, L. and Rokhlin, V. [1990]. On the Accurate Calculation of Vortex Shedding. *Phys. Fluids A* **2**, 883–885.

[11] Appel, A. W. [1985]. An Efficient Program for Many Body Simulations. *SIAM J. Sci. Stat. Comput.* **6**, 85–103.

[12] Ashurst, W. T. and Meiburg, E. [1988]. Three Dimensional Shear Layers Via Vortex Dynamics. *J. Fluid Mech.* **180**, 87–116.

[13] Ashurst, W. T. and McMurtry, P. A. [1989]. Flame Generation of Vorticity: Dipoles from Monopoles. *Combust. Sci. Tech.* **66**, 17–37.

[14] Baden, S. B. [1987]. Run Time Partitioning of Scientific Continuum Calculations Running on Multiprocessors. *Ph.D Thesis*, U. C. Berkeley Computer Science Dept.

[15] Baden, S. B. [1988]. Very Large Vortex Calculations in Two Dimensions. *Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lecture Notes in Mathematics **1360**, Springer Verlag, New York, 96–120.

[16] Baden, S. B. and Puckett, E. G. [1990]. A Fast Vortex Method for Computing 2D Viscous Flow. *J. Comp. Phys.* **91**, 278–297.

[17] Baker, G. R. [1979]. The "Cloud-in-Cell" Technique Applied to the Roll Up of Vortex Sheets. *J. Comp. Phys.* **31**, 76–95.

[18] Barnes, J. and Hut, P. [1986]. A Hierarchical $O(N \log N)$ Force Calculation Algorithm. *Nature* **324**, 446–449.

[19] Beale, J. T. [1986]. A Convergent 3-D Vortex Method with Grid-Free Stretching. *Math. Comp.* **46**, 401–424.

[20] Beale, J. T. [1988]. On the Accuracy of Vortex Methods at Large Times. *Computational Fluid Dynamics and Reacting Gas Flows* (Ed. by B. Engquist et al.), Springer Verlag, New York, 19–32.

[21] Beale, J. T., Eydeland, A., and Turkington, B. [1991]. Numerical Tests of 3-D Vortex Methods using a Vortex Ring with Swirl. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 1-9.

[22] Beale, J. T., Kato, T., and Majda, A. [1984]. Remarks on the Breakdown of Smooth Solutions for the 3-D Euler Equations. *Commun. Math. Phys.* **94**, 61–66.

[23] Beale, J. T. and Majda, A. [1981]. Rates of Convergence for Viscous Splitting of the Navier-Stokes Equations. *Math. Comp.* **37**, 243–259.

[24] Beale, J. T. and Majda, A. [1982a]. Vortex Methods I: Convergence in Three Dimensions. *Math. Comp.* **39**, 1–27.

[25] Beale, J. T. and Majda, A. [1982b]. Vortex Methods II: High Order Accuracy in Two and Three Dimensions. *Math. Comp.* **39**, 29–52.

[26] Beale, J. T. and Majda, A. [1984]. Vortex Methods for Fluid Flow in Two or Three Dimensions. *Contemp. Math.* **28**, 221–229.

[27] Beale, J. T. and Majda, A. [1985]. High Order Accurate Vortex Methods with Explicit Velocity Kernels. *J. Comp. Phys.* **58**, 188–208.

[28] Benfatto, G. and Pulvirenti, M. [1983]. A Diffusion Process Associated to the Prandtl Equations. *J. Funct. Anal.* **52**, 330–343.

[29] Birdsall, C. K. and Fuss, D. [1969]. Clouds-in-Clouds, Clouds-in-Cells, Physics for Many-Body Plasma Simulations. *J. Comp. Phys.* **3**, 494–511.

[30] Buneman, O. [1973]. Subgrid Resolution of Flows and Force Fields. *J. Comp. Phys.* **11**, 250–268.

[31] Buneman, O., Couet, B. and Leonard, A [1981]. Simulation of Three Dimensional Incompressible Flows with a Vortex-in-Cell Method. *J. Comp. Phys.* **39**, 305–328.

[32] Buttke, T. F. [1990]. A Fast Adaptive Vortex Method for Patches of Constant Vorticity in Two Dimensions. *J. Comp. Phys.* **89**, 161–186.

[33] Buttke, T. F. [1991]. Fast Vortex Method Methods in Three Dimensions. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 51–66.

[34] Caflisch, R. [1988]. Mathematical Analysis of Vortex Dynamics. *Mathematical Aspects of Vortex Dynamics* (Ed. by R. E. Caflisch), S.I.A.M., Philadelphia, 1–24.

[35] Carrier, J., Greengard, L. and Rokhlin, V. [1988] A Fast Adaptive Multipole Algorithm for Particle Simulations. *SIAM J. Sci. Stat. Comput.* **9**, 669–686.

[36] Chang, C. C. [1987]. Numerical Solution of Stochastic Differential Differential Equations with Constant Coefficients. *Math. Comp.* **49**, 523–542.

[37] Chang, C. C. [1988]. Random Vortex Methods for the Navier-Stokes Equations. *J. Comp. Phys.* **76**, 281–300.

[38] Cheer, A. Y. [1983]. A Study of Incompressible 2-D Vortex Flow Past a Circular Cylinder. *SIAM J. Sci. Stat. Comput.* **4**, 685–705.

[39] Cheer, A. Y. [1989]. Unsteady Separated Wake Behind an Impulsively Started Cylinder. *J. Fluid Mech.* **201**, 485–505.

[40] Chiu, C. and Nicolaides, R. A. [1988]. Convergence of a Higher-Order Vortex Method for Two-Dimensional Euler Equations. *Math. Comp.* **51**, 507–534.

[41] Choi, Y., Humphrey, J. A. C. and Sherman, F. S. [1988]. Random Vortex Simulation of Transient Wall-Driven Flow in a Rectangular Enclosure. *J. Comp. Phys.* **75**, 359–383.

[42] Choquin, J. P. and Lucquin-Desreux, B. [1988]. Accuracy of a Deterministic Particle Method for the Navier-Stokes Equations. *Inter. J. Num. Meth. Fluids* **8**, 1439–1458.

[43] Choquin, J. P. and Huberson, S. [1989]. Particle Simulations of Viscous Flows. *preprint* 27 pages.

[44] Chorin, A. J. [1973]. Numerical Study of Slightly Viscous Flow. *J. Fluid Mech.* **57**, 785–796.

[45] Chorin, A. J. [1978]. Vortex Sheet Approximation of Boundary Layers. *J. Comp. Phys.* **27**, 428–442.

[46] Chorin, A. J. [1980]. Vortex Models and Boundary Layer Instability. *SIAM J. Sci. Stat. Comput.* **1**, 1–21.

[47] Chorin, A. J. [1982]. The Evolution of a Turbulent Vortex. *Commun. Math. Phys.* **35**, 517–535.

[48] Chorin, A. J. and Bernard, P. S. [1973]. Discretization of a Vortex Sheet with an Example of Roll-up. *J. Comp. Phys.* **13**, 423–429.

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

[50] Christiansen, J. P. [1973]. Numerical Simulation of Hydrodynamics by the Method of Point Vortices. *J. Comp. Phys.* **13**, 363–379.

[51] Cottet, G. H. [1982]. Methodes Particulaires pour l'equation d'Euler dans le plan. *These de 3eme cycle*, l'Universite Pierre et Marie Curie, Paris VI.

[52] Cottet, G. H. [1987]. Convergence of a Vortex in Cell Method for the Two Dimensional Euler Equations. *Math. Comput.* **49**, 407–425.

[53] Cottet, G. H. [1988a]. Boundary Conditions and Deterministic Vortex Methods for the Navier-Stokes Equations. *Mathematical Aspects of Vortex Dynamics* (Ed. by R. E. Caflisch), S.I.A.M., Philadelphia, 128–143.

[54] Cottet, G. H. [1988b]. A New Approach to the Analysis of Vortex Methods in Two and Three Dimensions. *Ann. Inst. H. Poincare Anal. non Lineaire* **5**, 227–285.

[55] Cottet, G. H. [1990]. A Particle-Grid Superposition Method for the Navier-Stokes Equations. *J. Comp. Phys.* **89**, 301–318.

[56] Cottet, G. H. [1991]. Large Time Behaviour for Deterministic Particle Approximations to the Navier-Stokes Equations. *Math. Comput.* **56**, 45–60.

[57] Cottet, G. H., Goodman, J. and Hou, T. Y. [1991]. Convergence of the Grid-Free Point Vortex Method for the Three Dimensional Euler Equations. *SIAM J. Numer. Anal.* **28**, 291–307.

[58] Cottet, G. H. and Mas-Gallic, S. [1990]. A Particle Method to Solve the Navier-Stokes System. *Numer. Math.* **57**, 805–827.

[59] De Boor, C. [1978]. *A Practical Guide to Splines*, Springer-Verlag, New York.

[60] Ebin, D. and Marsden, J. E. [1970]. Groups of Diffeomorphisms and the Motion of an Incompressible Fluid. *Ann. of Math.* **92**, 102–163.

[61] Fishelov, D. [1990a]. Vortex Methods for Slightly Viscous Three Dimensional Flow. *SIAM J. Sci. Stat. Comput.* **11**, 399–424.

[62] Fishelov, D. [1990b]. A New Vortex Scheme for Viscous Flows. *J. Comp. Phys.* **86**, 211–224.

[63] Folland, G. B. [1976]. *A Introduction to Partial Differential Equations*, Princeton University Press, Princeton, New Jersey.

[64] Garabedian, P. R. [1986]. *Partial Differential Equations* Chelsea, New York.

[65] Ghoniem, A. F., Chorin, A. J. and Oppenheim, A. K. [1982]. Numerical Modelling of Turbulent Flow in a Combustion Tunnel. *Phil. Trans. R. Soc. Lond. A* **304**, 303–325.

[66] Ghoniem, A. F. and Sherman, F. S. [1985]. Grid Free Simulation of Diffusion Using Random Walk Methods. *J. Comp. Phys.* **61**, 1–37.

[67] Goodman, J. [1987]. Convergence of the Random Vortex Method. *Commun. Pure Appl. Math.* **40**, 189–220.

[68] Goodman, J., Hou, T. Y. and Lowengrub, J. [1990]. The Convergence of the Point Vortex Method for the 2-D Euler Equations. *Commun. Pure Appl. Math.* **43**, 415–430.

[69] Goodman, J. and Hou, T. Y. [1991]. New Stability Estimates for the 2-D Vortex Method. *Commun. Pure Appl. Math.* **44**, 1015–1031.

[70] Greenbaum, A., Greengard, L. and McFadden, G. B. [1991] Laplace's Equation and the Dirichlet-Neumann Map in Multiply Connected Domains. NYU Research Report, DOE/ER/25053-5 27 pages.

[71] Greengard, C. [1985]. The Core Spreading Vortex Method Approximates the Wrong Equation. *J. Comput. Phys.* **61**, 345–347.

[72] Greengard, C. [1986]. Convergence of the Vortex Filament Method. *Math. Comp.* **47**, 387–398.

[73] Greengard, L. [1988]. Potential Flow in Channels. *SIAM J. Sci. Stat. Comput.* **11**, 603–620.

[74] Greengard, L. [1990]. The Numerical Solution of the N-Body Problem. *Computers in Physics*, Mar/Apr, 142–152.

[75] Greengard, L. and Gropp, W. D. [1990]. A Parallel Version of the Fast Multipole Method. *Computers Math. Applic.* **20**, 63–71.

[76] Greengard, L. and Rokhlin, V. [1987]. A Fast Algorithm for Particle Simulations. *J. Comput. Phys.* **73**, 325–348.

[77] Greengard, L. and Rokhlin, V. [1988a] The Rapid Evaluation of Potential Fields in Three Dimensions. *Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lecture Notes in Mathematics **1360**, Springer Verlag, New York. 121–141.

[78] Greengard, L. and Rokhlin, V. [1988b] On the Efficient Implementation of the Fast Multipole Algorithm. Research Report, YALEDU/DCS/RR-602, 17 pages.

[79] Jolles, A. and Huberson, S. [1989]. Numerical Simulation of Navier-Stokes Equations by Means of Particle/ Mesh Methods. *LIMSI Research Report* **89-4**, 1–47.

[80] Hald, O. H. [1979]. Convergence of Vortex Methods for Euler's Equations II. *SIAM J. Numer. Anal.* **16**, 726–755.

[81] Hald, O. H. [1986]. Convergence of a Random Method with Creation of Vorticity. *SIAM J. Sci. Stat. Comput.* **7**, 1373–1386.

[82] Hald, O. H. [1987]. Convergence of Vortex Methods for Euler's Equations III. *SIAM J. Numer. Anal.* **24**, 538–582.

[83] Hald, O. H. [1991]. Convergence of Vortex Methods. *Vortex Methods and Vortex Motion* (Ed. by K. Gustafasson and J. Sethian), SIAM, Philadelphia, 33–58.

[84] Hald, O. H. and del Prete, V . M. [1978]. Convergence of Vortex Methods for Euler's Equations. *Math. Comp.* **32**, 791–809.

[85] Handscomb, D. C. and Hammersley, J. M. [1964]. Monte Carlo Methods. Methuen & Co. Ltd., London.

[86] Henshaw, W., Kreiss, H. O. and Reyna, L. G. [1991]. On Smallest Scale Estimates and a Comparison of the Vortex Method to the Pseudo-spectral Method. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 303–325.

[87] Hockney, R. W., Goel, S. P. and Eastwood, J. W. [1974]. Quiet High-Resolution Computer Models of a Plasma. *J. Comp. Phys.* **14**, 148–158.

[88] Hockney, R. W., and Eastwood, J. W. [1981]. *Computer Simulations Using Particles* McGraw Hill, New York.

[89] Hou, T. Y. [1992]. A New Desingularization Method for Vortex Methods. *Math. Comp.* (to appear).

[90] Hou, T. Y. [1991b]. A Survey on Convergence Analysis for Point Vortex Methods. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 327–339.

[91] Hou, T. Y. and Lowengrub, J. [1990]. Convergence of a Point Vortex Method for the 3-D Euler Equations. *Commun. Pure Appl. Math.* **43**, 965–981.

[92] Hou, T. Y., Lowengrub, J. and Shelly, M. [1991]. Exact Desingularization and Local Regridding for Vortex Methods. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 341–362.

[93] Hou, T. Y., and Wetton, B. T. R. [1990]. Convergence of a Finite Difference Scheme for the Navier-Stokes Equations using Vorticity Boundary Conditions.*SIAM J. Numer. Anal.* (to appear).

[94] Howell, L. H. [1990]. Computation of Conformal Maps by Modified Schwarz-Christoffel Transformations. *Ph.D Thesis*, MIT Math. Dept.

[95] Howell, L. H. and Trefethen, L. N. [1990]. A Modified Schwarz-Christoffel Transformation for Elongated Regions. *SIAM J. Sci. Stat. Comput.* **11**, 928–949.

[96] Kato, T. [1967] On Classical Solutions of the Two-Dimensional Non-Stationary Euler Equations. *Arch. Rat. Mech. Anal.* **25**, 188–200.

[97] Kato, T. [1972] Nonstationary Flows of Viscous and Ideal Fluids in R^3. *J. Funct. Anal.* **9**, 296–305.

[98] Knio, O. and Ghoniem, A. F. [1990]. Numerical Study of a Three Dimensional Vortex Method. *J. Comp. Phys.* **86**, 75–106.

[99] Krasny, R. [1986a]. Desingularization of Periodic Vortex Sheet Roll-up. *J. Comput. Phys.* **65**, 292–289.

[100] Krasny, R. [1986b]. A Study of Singularity Formation in a Vortex Sheet by the Point Vortex Approximation. *J. Fluid Mech.* **167**, 65–93.

[101] Krasny, R. [1987]. Computation of Vortex Sheet Roll-Up in the Trefftz Plane. J. Fluid Mech. **184**, 123–155.

[102] Kuwahara, K. and Takami, H. [1973] Numerical Studies of Two-Dimensional Vortex Motion by a System of Points J. Phys. Soc. Japan, **34**, 247–253

[103] Leonard, A. [1980]. Vortex Methods for Flow Simulation. *J. Comp. Phys.* **37**, 289–335.

[104] Leonard, A. [1985]. Computing Three Dimensional Flows with Vortex Elements. *Ann. Rev. Fluid Mech.* **17**, 523–559.

[105] Long, D. G. [1988]. Convergence of the Random Vortex Method in Two Dimensions. *J. A.M.S.* **4**, 779–804.

[106] Long, D. G. [1990]. Convergence of the Random Vortex Method in Three Dimensions. *Lawrence Berkeley Laboratory Report*, LBL-29832, 1–63.

[107] Lucquin-Desreux, B. [1987]. Particle Approximation of the Two-Dimensional Navier-Stokes Equations. *Rech. Aerosp.* **4**, 1–12.

[108] Majda, A. [1986] Vorticity and the Mathematical Theory of Incompressible Flow. *Commun. Pure Appl. Math.* **19**, (special volume) S187-S220.

[109] Majda, A. [1988] Vortex Dynamics: Numerical Analysis, Scientific Computing, and Mathematical Theory. *ICIAM '87: Proceedings of the First International Congress of Industrial and Applied Mathematics.* (Paris 1987) SIAM, 153–182.

[110] Maltz, F. H. and Hitzl, D. L. [1979]. Variance Reduction in Monte-Carlo Computations Using Multi-Dimensional Hermite Polynomials. *J. Comp. Phys.* **32**, 345–376.

[111] Marchioro, C. and Pulvirenti M. [1982] Hydrodynamics in Two Dimensions and Vortex Theory. *Commun. Math. Phys.* **84**, 483–503.

[112] Marchioro, C. and Pulvirenti M. [1983] A Diffusion Process Associated to the Prandtl Equation. *J. Func. Anal.* **52**, 330–343.

[113] Marsden, J. E. and Tromba, I. [1981]. *Vector Calculus*, Freeman and Co., San Francisco.

[114] Mas-Gallic, S. [1990]. Deterministic Particle Method: Diffusion and Boundary Conditions. *Universite Pierre et Marie Curie, Centre National de la Recherche Scientifique*, R90029, 1–27.

[115] Mayo, A. [1985]. Fast High Order Accurate Solution of Laplace's Equation on Irregular Domains. *SIAM J. Sci. Stat. Comput.* **6**, 144–157.

[116] McGrath, F. J. [1968] On Classical Solutions of the Two-Dimensional Non-Stationary Euler Equations. *Arch. Rational Mech. Anal.* **27**, 329–348.

[117] Merriman, B. [1991]. Particle Approximation. *Vortex Dynamics and Vortex Methods* (Ed. by C. Anderson and C. Greengard), Lectures in Applied Mathematics **28**, American Mathematical Society, Providence, RI, 481–546.

[118] Milinazzo, F. and Saffman, P. G. [1977]. The Calculation of Large Reynolds Number Two-Dimensional Flow Using Discrete Vortices with Random Walk. *J. Comp. Phys.* **23**, 380–392.

[119] Nicolaides, R. A. [1986]. Construction of a High Order Accurate Vortex and Particle Methods. *Appld. Numer. Math.* **2**, 313–320.

[120] Nordmark, H. O. [1988]. Higher Order Vortex Methods with Rezoning. *Ph.D Thesis*, U. C. Berkeley Math. Dept.

[121] Nordmark, H. O. [1991]. Rezoning for Higher Order Vortex Methods. *J. Comp. Phys.* **97**, 366–397.

[122] Perlman, M. [1985]. On the Accuracy of Vortex Methods. *J. Comp.Phys.* **59**, 200–223.

[123] Puckett, E. G. [1987]. A Study of Two Random Walk Methods and Their Rates of Convergence. *Ph.D Thesis*, U. C. Berkeley Math. Dept.

[124] Puckett, E. G. [1989a]. A Study of the Vortex Sheet Method and Its Rate of Convergence. *SIAM J. Sci. Stat. Comput.* **10**, 298–327.

[125] Puckett, E. G. [1989b]. Convergence of a Random Particle Method to Solutions of the Kolmogorov Equation $u_t = \nu u_{xx} + u(1-u)$. *Math. Comp.* **52**, 615–645.

[126] Raviart, P. A. [1985]. An Analysis of Particle Methods. *Numerical Methods in Fluid Dynamics* (Ed. by F. Brezzi), Lecture Notes in Mathematics **1127**, Springer Verlag, New York, 245–324.

[127] Quartapelle, L. [1981]. Vorticity Conditioning in the Computation of Two Dimensional Viscous Flows. *J. Comp. Phys.* **40**, 453–477.

[128] Quartapelle, L. and Valz-Griz, F. [1981]. Projection Conditions on the Vorticity in Viscous Incompressible Flows. *Int. J. Numer. Methods Fluids.* **1**, 129–144.

[129] Roberts, S. G. [1985]. Accuracy of the Random Vortex Methods for a Problem with Non-smooth Initial Conditions. *J. Comp. Phys.* **58**, 29–43.

[130] Roberts, S. G. [1989]. Convergence of a Random Walk Method for the Burgers Equation. *Math. Comp.* **52**, 647–673.

[131] Rokhlin, V [1985] Rapid Solution of Integral Equations of Classical Potential Theory. *J. Comput. Phys.* **60**, 187–207.

[132] Rosenhead, L. [1931]. The Formation of Vortices from a Surface of Discontinuity. *Proc. R. Soc. Lon., A* **134**, 170–192.

[133] Schlichting, H. [1979]. *Boundary-Layer Theory*, McGraw Hill, New York.

[134] Schultz, M. H. [1973]. *Spline Analysis*, Prentice-Hall Inc., Englewood Cliffs, NJ

[135] Sethian, J. A. [1984]. Turbulent Combustion in Open and Closed Vessels. *J. Comp. Phys.* **54**, 425–456.

[136] Sethian, J. A. [1991]. A Brief Overview of Vortex Methods. *Vortex Methods and Vortex Motion* (Ed. by K. Gustafasson and J. Sethian), SIAM, Philadelphia, 1–32.

[137] Sethian, J. A. and Ghoniem, A. F. [1988]. Validation Study of Vortex Methods. *J. Comp. Phys.* **74**, 283–317.

[138] Siggia, E. D. and Aref, H. [1981]. Point Vortex Simulation of the Inverse Energy Cascade in Two-Dimensional Turbulence. *Phys. Fluids* **24**, 171–173.

[139] Summers, D. M. [1989]. A Random Vortex Simulation of Falkner-Skan Boundary Layer Flow. *J. Comp. Phys.* **85**, 86–103.

[140] Summers, D. M., Hanson, T. and Wilson, C. B. [1985]. A Random Vortex Simulation of Wind Flow Over a Building. *Int. J. Num. Meth. Fluids* **5**, 849–871.

[141] van Dommelen, L. and Rundensteiner, E. A. [1989]. Fast Adaptive Summation of Point Forces in the Two Dimensional Poisson Equation. *J. Comp. Phys.* **83**, 126–147.

[142] Tennekes, H. and Lumley, J. L. [1972]. *A First Course in Turbulence*, MIT Press, Cambridge, MA.

[143] Tiemroth, E. [1986]. The Simulation of the Viscous Flow Around a Cylinder by the Random Vortex Method. *Ph.D Thesis*, U. C. Berkeley Naval Arch. Dept.

[144] White, F. M. [1974]. *Viscous Fluid Flow*, McGraw Hill, New York.

[145] Ying, L. A. [1990]. Viscous Splitting for the Unbounded Problem of the Navier-Stokes Equations. *Math. Comp.* **55**, 89–113.

[146] Zabusky, N. J., Hughes, M. H. and Roberts, K. V. [1979]. Contour Dynamics for the Euler Equations in Two Dimensions. *J. Comp. Phys.* **30**, 96–106.

[147] Zabusky, N. J., and Overman, E. A. [1983]. Regularization and Contour Dynamical Algorithms I. Tangential Regularization. *J. Comp. Phys.* **52**, 351–373.

[148] Zhu, J. [1989]. An Elliptical Vortex Method. *Ph.D Thesis*, NYU Courant Institute.