THE EFFECT OF VISCOUS DIFFUSION IN DISCRETE VORTEX DYNAMICS FOR SLIGHTLY VISCOUS FLOWS

By

Chien-Cheng Chang
and
Ruey-Ling Chern

IMA Preprint Series # 568
August 1989
The Effect of Viscous Diffusion in Discrete Vortex Dynamics for Slightly Viscous Flows

Chien-Cheng Chang *
and
Ruey-Ling Chern **

Institute of Applied Mechanics
National Taiwan University
Taipei, TAIWAN, R.O.C.

Abstract

Viscous vortex methods are obtained by incorporating the viscous diffusion effect into inviscid vortex methods in two and three space dimensions. We describe how the present methods are related to the inviscid methods, and compare their accuracy by a residual error analysis. Numerical results indicate that the methods converge rapidly up to a time inversely proportional to the kinematic viscosity. For long time calculations, a procedure for remeshing the vorticity field is proposed to ensure the accuracy of the inviscid and the proposed viscous methods. Efficient fast algorithms based on multipole expansions are employed to perform far-field calculations.

Key words: vortex methods, remesh, cutoff functions, viscous diffusion

1. Introduction

In this paper, we consider vortex methods for approximating solutions to the Navier-Stokes equations for incompressible flows,

\[
\frac{\partial}{\partial t} u + (u \cdot \nabla) u = -\frac{1}{\rho} \nabla p + \nu \Delta u ,
\]

(1.1)

\[
\nabla \cdot u = 0.
\]

(1.2)

In these equations, \( u = u(t, x) \) denotes the velocity, \( p = p(t, x) \) the pressure, \( \rho \) is the density, and \( \nu \) denotes the kinematic viscosity. We denote by \( N \) the number of space dimensions, which is either two or three.

* Associate Professor, supported in part by the NSC, Taiwan, R.O.C.
** Graduate Student, supported in part by the NSC, Taiwan, R.O.C.
During the past several years, vortex methods have been very attractive tools for numerical simulation of concentrated vortical flows, e.g., boundary layers and turbulence. The development of inviscid vortex methods has received a significant progress in both the design of methods and their theoretical analysis [1,2,3]. For slightly viscous flows, one may add appropriate Gaussian noise terms to the inviscid methods to account for the effect of viscous diffusion. The resultant random schemes have been used to study a number of physical problems and were able to produce nicely qualitative pictures of flow motions, e.g., see [7,13,20]. However, due to imperfect sampling, random schemes are not able to give accurate local flow information, although the evaluation of statistical quantities of flow fields can be fairly accurate by combining the use of high-order schemes and an efficient variance reduction technique [5,19].

Vortex methods involve the tracking of particle trajectories. Unlike the inviscid case, there do not exist exact characteristic formulations for the Navier-Stokes equations. It is therefore more difficult to obtain accurate vortex methods in the viscous case. However, by considering the physical effects in viscous flows, we are still able to obtain certain adequate approximate methods.

In this article, we propose to incorporate the effect of viscous diffusion into inviscid methods in a deterministic way. In particular, we consider the inviscid methods (T), (A) and (B) studied in [1]. The general procedure is as follows. The initial vorticity field is partitioned into a number of radially symmetric vortex blobs, each said to be positioned at the center of the blob. Due to the symmetry, a single vortex blob produces no net convective effect, but undergoes an isotropic diffusion. Meanwhile, the vortex blob is regarded, as a whole, to move with a velocity induced at the center of the blob due to the existence of other blobs. In three dimensions, due to vortex stretching, the associated vorticity of a vortex blob changes further according to the corresponding inviscid vorticity transport equations. We therefore generalize the methods (T'), (A') and (B') to the viscous case; and we will name by (T_\nu), (A_\nu) and (B_\nu) the corresponding viscous vortex methods.

The methods described above, especially in two dimensions, are not altogether new, and have been coined the term–core spreading vortex methods. Similar methods were proposed by Leonard in an earlier time [17], and which were shown by Greengard to approximate incorrect equations [14]. However, by concerning ourselves only with slightly viscous flows, and limiting the core spreading of vortex blobs, we are able to obtain adequate and useful numerical techniques. The way of limiting the core spreading is achieved
by adopting a remesh procedure to be described below. Above all, it is appropriate for us to investigate the physical background underlining the core spreading vortex methods.

The legitimacy of moving a vortex blob, as a whole, with a uniform velocity is due to its being highly concentrated in a small neighbourhood of its center. Hence, the cutoff size \( \sigma \) should be fairly small to ensure the validity of the approximation. However, according to the analysis of inviscid methods, the size of \( \sigma \) should not be less than the initial mesh size \( h \). Therefore, an appropriate choice of \( \sigma \) is an essential part in the design of (viscous) vortex methods. The introduction of \( \sigma \) signifies a difference from methods using point vortices in that the methods using vortex blobs may uniformly approximate accurately the flow field rather than in a weak sense.

Further justification of the above procedure can be made with an error analysis by investigating how the approximate flow quantities fail to satisfy the exact (viscous) vorticity transport equations. The analysis shows that the error form for the proposed viscous methods is completely the same as the corresponding inviscid methods. Accordingly, we have good reasons to believe that our methods can be as accurate as their inviscid counterparts for time \( t \) up to several \( \sigma^2/\nu \). At a time about the order of \( \sigma^2/\nu \), the vortex blobs have undergone substantial diffusion. However, we note that inviscid flows usually distort much faster than the corresponding viscous flows. In three dimensions, the vorticity field further encounters substantial stretching. In all the cases, we remesh the vorticity field to ensure the accuracy of the methods, i.e., the whole vorticity field is partitioned into a new set of vortex blobs on an appropriate regular grid before we go on to approximate the flow field.

Generally speaking, it is difficult to study slightly viscous flows numerically, due to that numerical diffusion often dominates the physical diffusion. One specific feature of our methods is that the viscosity often appears in a factor of Gaussian distribution, which forms a product with the vorticity. Therefore, the physical viscous effect will not be suppressed by the numerical diffusity. For this reason and from the above discussion, our methods are quite suitable for simulating concentrated vortical flows with small viscosity, or large Reynolds number flows when boundaries exist. Thereafter we will call our methods restricted core spreading methods.

Numerical tests have been done to verify the validity of the restricted core spreading vortex methods. In particular, we consider, in two dimensions, a viscous flow field with a radially symmetric initial vorticity field. Time integrations are performed by em-
ploying Runge-Kutta methods of second order while a fourth-order radially symmetric
cutoff functions is adopted. Numerical errors are analyzed in suitably defined norms; and
quite conclusive results support that the proposed viscous methods have the same order
of accuracy as the corresponding inviscid methods. Furthermore, the idea of remeshing
the vorticity field proves successful in improving the accuracy for long time calculations.
Especially, we mention that the fast algorithms based on multipole expansions [10,11] are
employed to evaluate the stream functions, and hence the velocity fields due to far-field
vortex blobs. The fast algorithms vastly speed up the calculation of vortex interactions.
Typically, when the number of vortex blobs exceeds a thousand, the run time for per-
forming one-step calculation using direct methods takes several minutes on a Vax-8600
machine, while for fast algorithms, it takes only a few seconds.

In the following sections, we describe how to obtain the viscous vortex methods \( T_\nu \),
\( A_\nu \) and \( B_\nu \), discuss cutoff functions, make an error analysis of vortex algorithms, and
carry out numerical examples to validate these methods. The idea of remesh and the fast
algorithms based on multipole expansions for facilitating the evaluation of interactions due
to far-field vortex blobs are also described.

2. Two-Dimensional Vortex Methods

In two dimensions, the vorticity function \( \omega = \omega(t,x) \) is defined by

\[
\omega = \partial_1 u_2 - \partial_2 u_1
\]  

where \( u = (u_1, u_2) \) and \( \partial_i \) denotes partial differentiation with respect to the \( i \)th space
variable. Thereafter, we will assume that the initial vorticity field \( \omega(0,x) \) is of compact
support. By taking the curl of Eq. (1.1), and using Eq. (1.2), we obtain

\[
\frac{\partial}{\partial t} \omega + (u \cdot \nabla) \omega = \nu \Delta \omega. 
\]  

Suppose first that the initial vorticity \( \omega(0,x) \) is radially symmetric,

\[
\omega(0,x) = \omega_0(|x|). 
\]  

We show that for this initial data, the solution to the diffusion equation

\[
\frac{\partial}{\partial t} \omega = \nu \Delta \omega. 
\]
is also the solution to Eq. (2.2). Let * denote convolution. The solution to Eq. (2.4) with initial data (2.3) is given by
\[
\omega(t, x) = (\mathcal{D}_t \ast \omega_0)(t, x),
\]
(2.5)
where \( \mathcal{D}_t = \mathcal{D}_t(x) \) is the diffusion kernel,
\[
\mathcal{D}_t(x) = \frac{1}{4\pi \nu t} \exp\left(-\frac{|x|^2}{4\nu t}\right).
\]
(2.6)
The function \( d_t(x) \) can be regarded as a Gaussian density of mean 0 and variance \( 2\nu t \). Indeed, one may show that \( \omega(t, x) \) given in (2.5) is also radially symmetric. This can be seen by considering \( \omega(t, Qx) \), where \( Q \) is an arbitrary \( 2 \times 2 \) orthogonal matrix. Using (2.3) and (2.6), we have
\[
\omega(t, Qx) = (\mathcal{D}_t \ast \omega_0)(t, Qx)
\]
\[
= \int \mathcal{D}_t(Qx - y) \omega_0(y) \, dy
\]
\[
= \int \mathcal{D}_t(Q(x - z)) \omega_0(Qz) |Q| \, dz, \quad (y = Qz)
\]
\[
= \int \mathcal{D}_t(x - z) \omega_0(z) \, dz = \omega(t, x).
\]
(2.7)
More generally, the above proof can be modified to show that if \( \omega(0, x) \) is radially symmetric about a point \( x_0 \) instead of the origin, then the solution \( \omega(t, x) \) to Eq. (2.4) is also radially symmetric about the point \( x_0 \).

Next, we show that for a radially symmetric vorticity field, the convection term in (2.2) vanishes. Evidently, it suffices to show that the velocity induced by (2.5) is perpendicular to the radial direction. For this purpose, let us invoke the stream function. From the incompressibility condition (1.2), it follows that there is a stream function \( \psi = \psi(t, x) \) such that
\[
\partial_2 \psi = u_1, \quad -\partial_1 \psi = u_2.
\]
(2.8)
Then
\[
\omega = \partial_1 u_2 - \partial_2 u_1 = -\Delta \psi.
\]
(2.9)
On account of the symmetry of \( \omega(t, x) \), we may seek a solution of the form \( \psi(t, x) = \psi(t, r), \ r = |x| \), to the above equation. Then Eq. (2.9) becomes
\[
\frac{1}{r} \frac{d}{dr} \left( r \frac{d\psi}{dr} \right) = -\omega.
\]
(2.10)
According to the definition (2.8), we have therefore,

\[
  u(t, x) = \frac{1}{2\pi} \frac{\langle -x_2, x_1 \rangle}{|x|^2} \cdot (-2\pi r\psi') = (K \cdot \Phi)(x),
\]

(2.11)

where \( \Phi = -2\pi r\psi' \). The function \( K = K(\mathbf{x}) \), as defined by the second equality, is called the velocity kernel, since the velocity field is given by the Biot-Savart law,

\[
  u(t, x) = (K \ast \omega)(t, x) = \int K(x - y) \omega(t, y) \, dy.
\]

(2.12)

For later use, we write \( \Phi \) in an explicit form,

\[
  \Phi(r) = 2\pi \cdot \int_0^r \omega(s) \, ds,
\]

(2.13)

where \( \omega \) is regarded as an radially symmetric function.

By (2.7) and (2.11), we have \((u \cdot \nabla)\omega = 0\). Therefore, we conclude that an initially radially symmetric vorticity field contributes nothing to nonlinear convection, but just undergoes an isotropic diffusion. This observation forms the basis for our construction of vortex methods for the Navier-Stokes equations.

First of all, we note that in vortex methods a great care is taken to ensure numerical stability by avoiding the singularity of the kernel \( K = K(\mathbf{x}) \) at the origin. A common practice is to convolve \( K \) with a smooth function \( f_\sigma \) of integral one, and then approximate the resultant velocity field,

\[
  K_\sigma \ast \omega = (K \ast f_\sigma) \ast \omega,
\]

(2.14)

where \( K_\sigma \), as defined by the equality, is close to \( K \) except at the origin, where \( K_\sigma \) is bounded. Furthermore, \( \sigma \) is called the cutoff size, and \( f_\sigma \) is called the cutoff function. It is important to note that, due to the associative law of convolution, (2.14) can also be written as

\[
  K_\sigma \ast \omega = K \ast (f_\sigma \ast \omega)
  = K \ast \int f_\sigma(\cdot - y) \omega(y) \, dy.
\]

(2.15)

Although there is no difference between (2.14) and (2.15), it is easier for us, in the present study, to approximate \( f_\sigma \ast \omega \) in (2.15), and then recover exactly the velocity field by a stream function formulation.
The basic idea of vortex methods is to track those particles which are initially located on a regular grid. We denote by $\Lambda_h$ the set of nodes of a regular grid with typical mesh width $h$, and which are contained in the support of the initial vorticity field. Let

$$\{\alpha_j = jh, j = (j_1, j_2)\}$$

denote the set of the nodes. For arbitrary initial vorticity field $\omega_0(x)$, we approximate the integral in (2.15) by a sum of 'vortex blobs',

$$\tilde{\omega}_0(x) = \sum_{j \in \Lambda_h} f_\phi(x - \alpha_j) p_j \omega_j h^2,$$  \hspace{1cm} (2.16)

where $\omega_j = \omega_0(\alpha_j)$, $p_j$ is the quadrature weight associated with the $j$th vortex blob. Furthermore, the cutoff function $f_\phi(x) = f_\phi(|x|)$ belongs to a class of radially symmetric functions, which will be specified later. Each component in (2.16) is called a vortex blob because $f_\phi$ is always chosen to be a bell-shaped function.

For the sake of comparison, we first describe the vortex method for inviscid flows with the initial vorticity field given by (2.16), and then proceed with the viscous case. For $\nu = 0$, the vorticity transport equation (2.2) becomes

$$\frac{\partial}{\partial t} \omega + (u \cdot \nabla) \omega = 0.$$  \hspace{1cm} (2.17)

In Lagrangean coordinates, the above equation can be written as

$$\frac{\partial}{\partial t} \omega(t, x(t, \alpha)) = 0, \quad x(0, \alpha) = \alpha.$$  \hspace{1cm} (2.18)

The physical meaning of Eq. (2.18), or equivalently (2.17), is then obvious. That is, the vorticity of a fluid particle does not change during the time evolution of a two-dimensional inviscid flow. This observation leads us to design a vortex algorithm which is equivalent to letting $\{\alpha_j\}$ move according to the induced velocity field. In terms of formulas, the vorticity field is given by

$$\tilde{\omega}(t, x) = \sum_{j \in \Lambda_h} f_\phi(x - \bar{x}_j(t)) p_j \omega_j h^2,$$  \hspace{1cm} (2.19)

while the vortex blobs evolve according to

$$\frac{d\bar{x}_i}{dt}(t) = \sum_{j \in \Lambda_h} (K \cdot \Phi)(\bar{x}_i(t) - \bar{x}_j(t)) p_j \omega_j h^2, \quad \bar{x}_i(0) = \alpha_i.$$  \hspace{1cm} (2.20)
where
\[ \Phi(r) = 2\pi \cdot \int_0^r \sigma(s) s \, ds. \] (2.21)

It is clear that \( \Phi \) also depends on the parameter \( \sigma \); however, we suppress this dependence for the sake of simplicity. In general, for arbitrary point \( x \) in the flow field at time \( t \), the approximate velocity field \( \tilde{u} = \tilde{u}(t, x) \) is determined from
\[ \tilde{u}(t, x) = \sum_{j \in A} (K \cdot \Phi)(x - \tilde{x}_j(t))p_j \omega_j h^2. \] (2.22)

Formulas (2.19)-(2.22) exactly correspond to the algorithm (T) studied in [1], except that here we use radially symmetric cutoff functions. The expressions (2.21) and (2.22) are derived from (2.19) in a similar way as (2.11) and (2.13) from (2.10), by seeking a stream function \( \tilde{\psi} \), for the \( \tilde{\omega} \) in (2.19), of the form \( \sum_{j \in A} \tilde{\psi}_j \) with \( r_j = |x - \tilde{x}_j(t)| \).

Next we formulate a vortex method for viscous flows (\( \nu \neq 0 \)). As time evolves, we obtain the approximate vorticity field \( \tilde{\omega}(t, x) \) by convolving \( D_t \) with \( \tilde{\omega}_0 \) according to (2.5), and letting each vortex blob move with the velocity induced at its center. In terms of formulas, we have
\[ \tilde{\omega}(t, x) = \sum_{j \in A} (D_t \ast \sigma)(x - \tilde{x}_j(t))p_j \omega_j h^2, \] (2.23)

while the vortex blobs evolve according to
\[ \frac{d\tilde{x}_j(t)}{dt} = \sum_{j \in A} (K \cdot \Phi)(\tilde{x}_j(t) - \tilde{x}_j(t))p_j \omega_j h^2, \quad \tilde{x}_j(0) = \alpha_j, \] (2.24)

where
\[ \Phi(r) = 2\pi \cdot \int_0^r (D_t \ast \sigma) s \, ds. \] (2.25)

We observe that \( \Phi \) is further a function of time \( t \) in the viscous case. The general velocity field is, again, determined from (2.22), but with \( \Phi \) given by (2.25) for the viscous case. Note that (2.20) and (2.24) are completely the same in form.

In summary, we have the following inviscid algorithm by collecting (2.20) and (2.21).
Algorithm \((T)\)

\[
\tilde{x}_i(0) = \alpha_i,
\]

\[
\frac{d\tilde{x}_i}{dt}(t) = \sum_{j \in \Lambda} (K \cdot \Phi)(\tilde{x}_i(t) - \tilde{x}_j(t))p_j \omega_j h^2,
\]

\[
\Phi(r) = 2\pi \cdot \int_0^r f_\sigma s \, ds.
\]

And by collecting (2.24) and (2.25), we have the following viscous algorithm.

Algorithm \((T_\nu)\)

\[
\tilde{x}_i(0) = \alpha_i,
\]

\[
\frac{d\tilde{x}_i}{dt}(t) = \sum_{j \in \Lambda} (K \cdot \Phi)(\tilde{x}_i(t) - \tilde{x}_j(t))p_j \omega_j h^2,
\]

\[
\Phi(r) = 2\pi \cdot \int_0^r (D_t \ast f_\sigma) s \, ds.
\]

The physical meaning of Algorithm \((T_\nu)\) is that besides undergoing an isotropic diffusion, each vortex blob is regarded, as a whole, to move with a velocity due to the existence of other vortex blobs. Intuitively, the approximation should be valid whenever each vortex blob is still concentrated in a relatively small region. However, in a long period, a vortex blob may have undergone substantial diffusion, it is no longer valid to assume that the whole blob moves with a uniform velocity.

To overcome the above difficulty, a remesh procedure is adopted. That is, we partition the vorticity field into a new set of vortex blobs on a regular grid for a period of the order \(\sigma^2/\nu\). This procedure ensures the accuracy of the viscous algorithm \((T_\nu)\) for long time calculations. Further justification based on a residual error analysis will be carried out in a later section.

We now discuss how the algorithm \((T_\nu)\) is related to the inviscid algorithm \((T)\). The difference between these two algorithms is reflected in (2.19) and (2.23), and (2.21) and (2.25), where \(D_t = D_t(x)\) is replaced by the Dirac delta function \(\delta = \delta(x)\) in the inviscid formulation. This is quite reasonable, since as the viscosity \(\nu\) tends to zero, the diffusion kernel behaves like a Dirac delta function. Actually, from our derivation of \((T_\nu)\), we observe that the viscous diffusion effect comes into play only via the convolution of \(D_t\) with
3. Three-Dimensional Vortex Methods

In three dimensions, the vorticity function \( \omega = \omega(t, x) \) is defined by

\[
\omega = \nabla \times u
\]  
(3.1)

and the vorticity transport equation becomes a bit complicated,

\[
\frac{\partial}{\partial t} \omega + (u \cdot \nabla) \omega = (\omega \cdot \nabla)u + \nu \Delta \omega.
\]  
(3.2)

There is a basic difference from the two-dimensional case. Instead of a scalar stream function, we have to introduce a vector stream function \( \Psi = \Psi(t, x) \) such that

\[
\Delta \Psi = -\omega,
\]  
(3.3)

and \( u \) is determined from

\[
u = \nabla \times \Psi.
\]  
(3.4)

In parallel to the previous discussion in two dimensions, we notice that the solution kernel \( D_t = D_t(x) \) associated with the three-dimensional diffusion equation

\[
\frac{\partial}{\partial t} \omega(t, x) = \nu \Delta \omega
\]  
(3.5)

is given by

\[
D_t(x) = \frac{1}{(4\pi \nu t)^{3/2}} \exp(-|x|^2/4\nu t).
\]  
(3.6)

Furthermore, for any given initial function \( \omega_0(x) \) which is radially symmetric about \( x_0 \), the solution to Eq. (3.5), given by \( \omega(t, x) = (D_t * \omega_0)(x) \), is also symmetric about the point \( x_0 \). Next we show how to obtain solutions to flow equations for radially symmetric vorticity functions. As we did in two dimensions, we seek a solution of the form \( \Psi = \Psi(t, r) \) to Eq. (3.3) for a radially symmetric function \( \omega = \omega(t, r) \),

\[
\frac{1}{r^2} \frac{d}{dr} \left( r^2 \frac{d}{dr} \Psi \right) = -\omega
\]  
(3.7)
Then by the formula (3.4), the velocity field $u = u(t,x)$ is given by

$$u(t,x) = -K \cdot 4\pi r^2 \Psi' = K \cdot \Phi,$$  

(3.8)

where $\Phi = -4\pi r^2 \Psi'(r)$. For later use, we write $\Phi$ in an explicit form,

$$\Phi(r) = 4\pi \cdot \int_0^r \varpi s^2 ds.$$  

(3.9)

Furthermore, in (3.8), $K = K(x)$ denotes the three-dimensional velocity kernel, since for arbitrary vorticity function $\varpi = \varpi(t,x)$ the velocity field is determined from $\varpi$ by the Biot-Savart law

$$u(t,x) = (K * \varpi)(t,x) = \int K(x - y) \varpi(t,y) dy.$$  

(3.10)

In matrix form, $K$ is expressed by

$$K(x) = \frac{1}{4\pi} \begin{pmatrix} 0 & \frac{-x_1}{|x|^3} & \frac{-x_2}{|x|^3} \\ \frac{-x_1}{|x|^3} & \frac{-x_2}{|x|^3} & \frac{-x_3}{|x|^3} \\ \frac{-x_3}{|x|^3} & \frac{-x_3}{|x|^3} & 0 \end{pmatrix}.$$  

(3.11)

From the above expression, it is clear that the function $K = K(x)$ has a singularity at the origin. In vortex methods, a care is taken to remove the singularity, as in two dimensions, by convolving $K$ with a smooth functions $f_\sigma = f_\sigma(x)$ of integral one,

$$K_\sigma * \varpi = K * (f_\sigma * \varpi) = K * \int f_\sigma(\cdot - y) \varpi(y) dy.$$  

(3.12)

One may easily verify from (3.8) that the velocity $u$ is perpendicular to the direction $(x_1, x_2, x_3)$; and therefore we conclude that the convection term vanishes, $(u \cdot \nabla) \varpi = 0$, for a radially symmetric vorticity function $\varpi$. However, due to the existence of the stretching term in three dimensions, we do not have a simple solution to the transport equation (3.2) for an initially radially symmetric vorticity function $\varpi_0(x)$.

Therefore, before we can do anything meaningful to formulate vortex methods in three dimensions, it is appropriate for us to discuss the effect of vortex stretching. In view of that
a rigorous theory for inviscid vortex methods has been established, we will first explain the idea underlying inviscid vortex methods that treat the effect of vortex stretching, and then derive vortex methods for viscous flows. For inviscid flows ($\nu = 0$), the vorticity transport equation (3.2) reduces to

$$\frac{\partial}{\partial t} \omega + (u \cdot \nabla)\omega = (\omega \cdot \nabla)u. \quad (3.13)$$

In Lagrangean coordinates, Eq. (3.13) is expressed by

$$\frac{\partial}{\partial t} \omega(t, x(t, \alpha)) = [\omega(t, x(t, \alpha)) \cdot \nabla_x] u(t, x(t, \alpha)) \quad (3.14)$$

where

$$x(0, \alpha) = \alpha. \quad (3.15)$$

Furthermore, there is another useful way to express the effect of vortex stretching without involving any derivatives of $\omega(t, x)$. In can be shown, e.g. in [10], that (3.14) is equivalent to

$$\omega(t, x(t, \alpha)) = [\nabla_x x(t, \alpha)] \cdot \omega_0(\alpha), \quad (3.16)$$

in which the effect of vortex stretching is reflected in the differential gradient of the flow map $x(t, \alpha)$. Formulas (3.14) and (3.16) constitute the basis for treating the effect of vortex stretching in vortex methods for both inviscid and viscous flows.

To formulate vortex methods, we need to track those particles which are initially located on a regular grid. We denote by $\Lambda_h$ the set of nodes of a regular grid with typical mesh width $h$, and which are contained in the support of the initial vorticity field. Let

$$\{\alpha_j = jh, j = (j_1, j_2, j_3)\}$$

denote the set of the nodes. For arbitrary initial vorticity field $\omega_0(x)$, we approximate the integral in (3.10) by a sum of vortex blobs,

$$\tilde{\omega}_0(x) = \sum_{\Lambda_h \in A_h} f_\sigma(x - \alpha_j) \omega_j p_j h^3, \quad (3.17)$$

where $\omega_j = \omega_0(\alpha_j)$, $p_j$ is the quadrature weight associated with the $j$th vortex blob. Furthermore, as in two dimensions, the cutoff function $f_\sigma(x) = f_\sigma(|x|)$ is always chosen to be bell-shaped, and belongs to a class of radially symmetric functions, which will be specified later.
Note that (3.17) is merely an approximation to the cutoff vorticity \( f_\sigma \ast \omega_0 \), and is not necessarily divergence-free. Below we will see that the effect of cutoff leads to smoothing the velocity kernel. Formulas (3.14) and (3.16) suggest two ways of evolving the vorticity field, and thus two ways of formulating vortex methods. In the following, we formulate vortex methods for inviscid flows and viscous flows separately. Their connections will then be described.

**Inviscid Algorithms.** The algorithms are based on the evolution equation (3.14) or (3.16) with the initial vorticity field approximated by (3.17). The resultant schemes are equivalent to letting \( \{ \alpha_j \} \) move according to the induced velocity field, and \( \{ \omega_0 (\alpha_j) \} \) undergo stretching according to the equation (3.14) or (3.16). In terms of formulas, the approximate vorticity field is given by

\[
\omega(t,x) = \sum_{Jh \in A_h} f_\sigma (x - \bar{x}_j(t)) \bar{\omega}_j(t) p_j h^3,
\]

(3.18)

where \( \bar{\omega}_j(0) = \omega_j \) while the vortex blobs evolve according to

\[
\frac{d\bar{x}_1}{dt}(t) = \sum_{Jh \in A_h} (K \cdot \Phi)(\bar{x}_1(t) - \bar{x}_j(t)) \bar{\omega}_j(t) p_j h^3,
\]

(3.19)

with \( \bar{x}_1(0) = \alpha_1 \). Notice that \( \{ \bar{\omega}_j(t) \} \) is different from \( \bar{\omega}(t,x) \) at \( x = x_j \), and which evolve according to either (3.14) or (3.16),

\[
\frac{d\bar{\omega}_i}{dt}(t) = \bar{\omega}_i(t) \cdot \sum_{Jh \in A_h} \nabla_x (K \cdot \Phi)(\bar{x}_1(t) - \bar{x}_j(t)) \bar{\omega}_j(t) p_j h^3,
\]

(3.20)

or,

\[
\bar{\omega}_i(t) = [\nabla^h \bar{x}(t, \alpha_i)] \cdot \omega_0 (\alpha_i).
\]

(3.21)

In the last formula, \( \nabla^h \) is a finite difference operator, which approximates the gradient \( \nabla \) in (3.16). The function \( \Phi \) is radially symmetric, and is given by

\[
\Phi(r) = 4\pi \cdot \int_0^r f_\sigma s^2 ds.
\]

(3.22)

More generally, for arbitrary point \( x \) in the flow field at time \( t \), the approximate velocity field \( \bar{u} = \bar{u}(t,x) \) is determined from the formula

\[
\bar{u}(t,x) = \sum_{Jh \in A_h} (K \cdot \Phi)(x - \bar{x}_j(t)) \bar{\omega}_j(t) p_j h^3.
\]

(3.23)
The expressions (3.22) and (3.23) are derived from (3.18) in a similar way as (3.8) and (3.10) from (3.7), by seeking a stream function \( \tilde{\psi} \), for the \( \tilde{\omega} \) in (3.18), of the form \( \sum_{j \in A_k} \tilde{\psi}_j(r_j) \) with \( r_j = |x - \tilde{x}_j(t)| \). Formulas (3.19) and (3.20) exactly correspond to Algorithm (A) studied in [1], while (3.19) and (3.21) correspond to Algorithm (B), except that here we use radially symmetric cutoff functions.

**Viscous Algorithms.** Next we formulate vortex methods for viscous flows (\( \nu \neq 0 \)). As time evolves, we obtain the approximate vorticity field \( \tilde{\omega}(t, x) \) by convolving \( D_t \) with \( \tilde{\omega}_0 \), and letting each vortex blob move with the velocity induced at its center, and \( \{ \tilde{\omega}_j \} \) undergo stretching according to either (3.14) or (3.16). In terms of formulas, we have

\[
\tilde{\omega}(t, x) = \sum_{j \in A_k} (D_t * f_\nu)(x - \tilde{x}_j(t)) \tilde{\omega}_j(t) p_j h^3,
\]  
(3.24)

and the vortex blobs evolve according to

\[
\frac{d\tilde{x}_j}{dt}(t) = \sum_{j \in A_k} (K \cdot \Phi)(\tilde{x}_j(t) - \tilde{x}_j(t)) \tilde{\omega}_j(t) p_j h^3,
\]  
(3.25)

with \( x_j(0) = \alpha_j \). The functions \( \{ \tilde{\omega}_j(t) \} \) evolve according to

\[
\frac{d\tilde{\omega}_j}{dt}(t) = \tilde{\omega}_j(t) \cdot \sum_{j \in A_k} \nabla_x (K \cdot \Phi)(\tilde{x}_j(t) - \tilde{x}_j(t)) \tilde{\omega}_j(t) p_j h^3,
\]  
(3.26)

or,

\[
\tilde{\omega}_j(t) = [\nabla^h_{\alpha} \tilde{x}(t, \alpha_j)] \cdot \tilde{\omega}_0(\alpha_j).
\]  
(3.27)

The function \( \Phi \) is given explicitly by

\[
\Phi(r) = 4\pi \cdot \int_0^r (D_t * f_\nu(s^2) \, ds.
\]  
(3.28)

We notice that for viscous flows, the general velocity field is, again, determined from (3.23), but with \( \Phi \) given by (3.28). Furthermore, (3.19) and (3.25) are completely the same in form, so are (3.20) and (3.26), or (3.21) and (3.27). To facilitate our discussions, we summarize by displaying the vortex methods which we have described in this subsection. Following [1], we denote by (A) and (B) the inviscid algorithms, while by (A_\nu) and (B_\nu) the viscous algorithms.

By collecting (3.19), (3.20) and (3.22), and (3.25), (3.26) and (3.28), we have
Algorithms (A), (A\_\nu)

\[ \ddot{X}_i(0) = \alpha_i, \]

\[ \frac{d\ddot{X}_i}{dt}(t) = \sum_{j \in \mathcal{L}_h} (K \cdot \Phi)(\ddot{X}_i(t) - \ddot{X}_j(t))\ddot{\psi}_j(t)p_j h^3, \]

\[ \frac{d\ddot{\psi}_i}{dt}(t) = \ddot{\psi}_i(t) \cdot \sum_{j \in \mathcal{L}_h} \nabla_x(K \cdot \Phi)(\ddot{X}_i(t) - \ddot{X}_j(t))\ddot{\psi}_j(t)p_j h^3, \]

\[ \Phi(r) = 4\pi \cdot \int_0^r f_\sigma s^2 ds, \ (\nu = 0) \ \text{or}, \]

\[ \Phi(r) = 4\pi \cdot \int_0^r (r_\sigma * f_\sigma) s^2 ds, \ (\nu > 0) \]

By collecting (3.19), (3.21) and (3.22), and (3.25), (3.27) and (3.28), we have

Algorithms (B), (B\_\nu)

\[ \ddot{X}_i(0) = \alpha_i, \]

\[ \frac{d\ddot{X}_i}{dt}(t) = \sum_{j \in \mathcal{L}_h} (K \cdot \Phi)(\ddot{X}_i(t) - \ddot{X}_j(t))\ddot{\psi}_j(t)p_j h^3, \]

\[ \ddot{\psi}_i(t) = [\nabla_\sigma x(t, \alpha_1)] \cdot \omega_\theta(\alpha_1), \]

\[ \Phi(r) = 4\pi \cdot \int_0^r f_\sigma s^2 ds, \ (\nu = 0) \ \text{or}, \]

\[ \Phi(r) = 4\pi \cdot \int_0^r (r_\sigma * f_\sigma) s^2 ds, \ (\nu > 0) \]

The physical meaning of the algorithms (A\_\nu) and (B\_\nu) is that besides undergoing an isotropic diffusion, each vortex blob is regarded, as a whole, to move with a velocity due to the existence of other vortex blobs, while the associated vorticity is subject to nonlinear stretching. Intuitively, the approximation should be valid whenever each vortex blob is still concentrated in a relatively small region.

For long time calculations, a remesh procedure, as indicated in the previous section, has to be adopted to ensure the accuracy of the vortex methods. We note that in three dimensions, the vorticity field may also encounter significant stretching, which also requires a careful treatment of grid. Further justification of the validity of the viscous algorithms (A\_\nu) and (B\_\nu) based on a residual error analysis will be carried out in the next section.
Now we discuss how the algorithms \((A_\nu)\) and \((B_\nu)\) are related to the inviscid algorithm \((A)\) and \((B)\). The only difference between these algorithms is reflected in (3.18) and (3.24), or (3.22) and (3.28), where \(D_t = D_t(x)\) is replaced by the Dirac delta function \(\delta = \delta(x)\) in the inviscid formulation. This is quite reasonable, since as the viscosity \(\nu\) tends to zero, the diffusion kernel behaves like a Dirac delta function. Actually, from our derivation of \((A_\nu)\) and \((B_\nu)\) we see that the viscous effect comes into play only via the convolution of \(D_t\) with \(f_\sigma\) through (3.24). Again, as in two dimensions, if we set \(\delta\) instead of \(D_t\), we will have a derivation of an inviscid method.

4. Cutoff Functions and Error Analysis

In this section we describe how to obtain cutoff functions which are particularly useful in the present study and employ them to investigate the accuracy of vortex algorithms.

Cutoff functions. We recall that a cutoff function \(f_\sigma = f_\sigma(x)\) is used to smooth the velocity kernel \(K = K(x)\) near the origin. A common practice to obtain a cutoff function is to set \(f_\sigma(x) = 1/\sigma^N f(x/\sigma)\), where \(f = f(x)\) is a smooth function of integral one. In order to get vortex methods of high order accuracy, a cutoff function has to fulfill further properties. Following [1], we require that \(f\) be a rapidly decaying function, satisfying the following conditions:

(i): \(\int_{R^N} f(x) \, dx = 1\),

(ii): \(\int_{R^N} x^\alpha f(x) \, dx = 0\), for all multi-indices \(\alpha\) such that \(1 \leq |\alpha| \leq p - 1\),

(iii): \(\int_{R^N} |x|^p |f(x)| \, dx < \infty\),

(iv): \(f \in C^L(R^N)\).

These conditions ensure the stability and accuracy of vortex methods for appropriate choices of the cutoff size \(\delta\) and the mesh size \(h\). For details, see [1,2,3]. One typical result from the analysis of inviscid methods is that for \(N = 3\), under suitable conditions, we have

\[
||\ddot{x}(t) - x(t)||_h \leq \left( \sigma^p + \left( \frac{h}{\sigma} \right)^L \sigma + h^r \right) \quad (4.1)
\]

where the norm \(|| \cdot ||_h\) is defined by

\[
||x(t)||_h = \left( \sum_{j \in \Lambda_h} |x_j(t)|^2 h^N \right)^{1/2}. \quad (4.2)
\]

Therefore, according to this estimate, for a given \(h\), the cutoff size \(\sigma\) cannot be arbitrarily small, but limited by the size of \(h\). This is reasonable because for very small \(\delta\), we have
approximation to the vorticity field by point vortex blobs, as can be seen from (2.19) or (3.18), and which is not able to uniformly approximate the vorticity field accurately. By high order cutoff functions we mean that $p$ is large.

Beale and Majda [4] have constructed radially symmetric cutoff functions which satisfy the conditions (i)-(iv), such that (2.21) (in two dimensions), or (3.22) (in three dimensions) can be evaluated explicitly. However, for viscous methods, we require that the convolutions in (2.25) (in two dimensions) and (3.28) (in three dimensions) be evaluated explicitly. Following the methodology in [4], we may easily find suitable cutoff functions for both inviscid and viscous algorithms. We list some useful cutoff functions $f(r)$ and their corresponding $\Phi(r)$ in Tables 1 and 2.

In Table 2, $\text{erf} = \text{erf}(x)$ denotes the error function, and which is defined by

$$\text{erf}(r) = \frac{2}{\sqrt{\pi}} \int_0^r e^{-s^2} ds. \quad (4.3)$$

We observe that the cutoff functions listed in Tables 1 and 2 are linear combinations of functions of Gaussian distribution. We will call them functions of Gaussian type. These functions can be readily convolved with the diffusion kernels to yield explicit formulas for $(D_t * f_\sigma)$ in two or three dimensions. Furthermore, we observe that each $\Phi$ is zero at the origin, and the order of zeroes is two in two dimensions, and three in three dimensions. The singularity of the kernel $K = K(x)$ at the origin (of order two in two dimensions, of order three in three dimensions) is therefore removed through the product $(K \cdot \Phi)(x)$.

Moreover, we note from Tables 1,2 that it is not altogether right to expand $\sigma^2$ at a rate of $2 \nu$, like a pure Gaussian process. Such a rate would only be correct for specific choices of Gaussian cutoff functions. Especially important for later use, we observe that all the functions $\Phi$'s in Tables 1 and 2 are rapidly tending to one when $r$ is sufficiently large. Indeed, we show that for all choices of cutoff functions $f$, the corresponding function $\Phi$ has the value one at infinity. We recall that $\Phi$ is also a function of time $t$. Define $J(t) = \Phi(r)|_{r \to \infty}$. Then the constancy of $J(t) \equiv 1$ for all $t \geq 0$ follows form the facts: (i) $J(0) = 1$ and (ii): $(D_t * f_\sigma)(x)$ is a solution to the diffusion equation (2.4), or (3.5) for all $f_\sigma$,

$$\frac{dJ}{dt}(t) = \int_{R^2} \frac{\partial}{\partial t} (D_t * f_\sigma) \, dx$$

$$= \int_{R^2} \nu \Delta_x (D_t * f_\sigma) \, dx$$
\[\nu \cdot \lim_{a \to \infty} \int_{|x| = a} \frac{\partial}{\partial n} \left( P_t \ast f_\sigma \right) dS = 0. \] (4.4)

The last two equalities follow from the rapidly decaying property of \((P_t \ast f_\sigma)(x)\) at infinity. Therefore, for \(r\) sufficiently large, we may practically take \(\Phi(r)\) to be one. This observation enables us to employ the fast algorithms based on mutipole expansions developed in [10,11] to facilitate the evaluation of velocities of vortex blobs due to far-field vortex interactions when performing the algorithms developed in the previous sections. We will come back to this subject in Section 5.

**Error Analysis.** For a rigorous error analysis of the proposed visocous methods, it may require delicate mathematical techniques. Instead, we analyze residual errors of the transport equations. In other words, we consider the accuracy of vortex methods by investigating how the approximate flow quantities fail to satisfy the vorticity transport equation (at centers of vortex blobs). In three dimensions, we analyze

\[ E_i(t) = \left\{ \frac{\partial}{\partial t} \tilde{\omega} + (\bar{u} \cdot \nabla) \tilde{\omega} - (\bar{\omega} \cdot \nabla) \bar{u} - \nu \Delta \bar{\omega} \right\}_i, \] (4.5)

where \(\tilde{\omega}\) is given by (3.24), and \(\bar{u}\) is given by (3.23). The expression (4.5) reduces, in two dimensions, to

\[ E_i(t) = \left\{ \frac{\partial}{\partial t} \tilde{\omega} + (\bar{u} \cdot \nabla) \tilde{\omega} - \nu \Delta \bar{\omega} \right\}_1, \] (4.6)

where \(\tilde{\omega}\) is given by (2.23), and \(\bar{u}\) is given by (2.22). The analysis is surely not restricted to vortex blobs' centers, but it is easy to look at the evolution of residual errors by (4.5), or (4.6). In the following derivation, the cutoff functions are assumed to be radially symmetric. For the sake of saving notations, we define

\[ \tilde{r}_{i,j}(t) = \tilde{x}_i(t) - \tilde{x}_j(t), \quad \bar{u}_{i,j}(t) = \bar{u}_i(t) - \bar{u}_j(t). \] (4.7)

After laborious calculations, we obtain in two dimensions,

\[ E_i(t) = \sum_{Jh \in \Lambda_h} \tilde{u}_{i,j}(t) \cdot \frac{\tilde{r}_{i,j}(t)}{|\tilde{r}_{i,j}(t)|} \left( P_t \ast f_\sigma \right)'(\tilde{r}_{i,j}(t)) p_h \omega_j h^2, \] (4.8)

and in three dimensions,

\[ E_i(t) = \sum_{Jh \in \Lambda_h} \tilde{u}_{i,j}(t) \cdot \frac{\tilde{r}_{i,j}(t)}{|\tilde{r}_{i,j}(t)|} \left( P_t \ast f_\sigma \right)'(\tilde{r}_{i,j}(t)) \bar{\omega}_j(t) p_j h^3 \]

\[ + \sum_{Jh \in \Lambda_h} \left[(\tilde{\omega}_j(t) \cdot \nabla) \bar{u}_{i,j}(t)\right] \left( P_t \ast f_\sigma \right)(\tilde{r}_{i,j}(t)) p_j h^3. \] (4.9)
A careful observation reveals that these error formulas are valid for both viscous and inviscid algorithms (by just letting $\nu \to 0$, or equivalently, by replacing $D_t(x)$ by $\delta(x)$). We notice that in these formulas, the first error term on the right hand side is due to convection, and the second error term (only in three dimensions) is due to vortex stretching.

No single error term is solely due to viscous diffusion in that the effect of viscous diffusion is taken into consideration by convolving with the diffusion kernels. In other words, in Algorithms (A$_\nu$) and (B$_\nu$), the evolution of vorticity due to viscous diffusion is exactly treated, while the error terms for both inviscid and viscous algorithms are completely the same in form.

To take a close look at the effect of viscous diffusion, we substitute a cutoff function from Table 1 or 2 into the above error formulas. For example, for $N = 2$, we use the fourth-order cutoff function of Gaussian type from Table 1, then

$$E_1(t) = \sum_{j_h \in \Lambda_h} \left\{ \frac{4}{\pi \sigma_1(t)^4} \exp \left( -\frac{\left| \tilde{r}_{i,j}(t) \right|^2}{\sigma_1(t)^2} \right) \right. $$

$$- \left. \frac{1}{2\pi \sigma_2(t)^4} \exp \left( -\frac{\left| \tilde{r}_{i,j}(t) \right|^2}{2\sigma_2(t)^2} \right) \right\} \times \tilde{u}_{i,j}(t) \cdot \tilde{r}_{i,j}(t) p_j \omega_j h^2 \right\}. \quad (4.10)$$

where

$$\sigma_1(t)^2 = \sigma^2 + 4 \nu t \quad \text{and} \quad \sigma_2(t)^2 = \sigma^2 + 2 \nu t.$$

Since there exists rigorous analysis for the inviscid algorithms (T), (A) and (B), the above error form may give us several implications. First, for high-order cutoff functions, the error $E_i$ is fairly small in case of $\nu = 0$, provided that $\sigma$ is appropriately chosen. Therefore, if $\sigma_1(t)$ and $\sigma_2(t)$ remain small, then one may expect that $E_i(t)$ is also small. In other words, before the vorticity field undergoes significant diffusion, the accuracy of viscous algorithms would be satisfactory. Second, in a long time calculation, it is worthwhile to partition the flow field on a new appropriate regular grid for a period of the order of several $\sigma^2/\nu$, as mentioned in Sections 2 and 3. This is the remesh procedure mentioned in previous sections, which is proposed to ensure the accuracy of viscous vortex methods in long time calculations; we shall come back to this subject in the next section.
5. Efficient Implementation of Vortex Algorithms

In this section we consider how to implement efficiently the algorithms developed in the previous sections, which include Algorithms (T), (Tν), (A), (Aν) and (B), (Bν). Our main concerns consist of fast computing velocities of vortex blobs and a procedure for ensuring the accuracy of vortex methods in long time calculations. These two aspects are actually intimately related to each other, and which will be described in the following contexts.

Fast algorithms. In implementing vortex algorithms, the main computational cost comes from the evaluation of interactions among vortex blobs. Consider a field of \( \mathcal{N} \) vortex blobs, it requires \( 1/2 \mathcal{N}(\mathcal{N} - 1) \) direct evaluations of vortex interactions, e.g.,

\[
(K \cdot \Phi)(\vec{x}_i(t) - \vec{x}_j(t)), \quad 1 \leq i, j \leq \mathcal{N},
\]

(5.1)

to obtain the velocities of all the vortex blobs. The cost would be exceedingly expensive for a moderately large number \( \mathcal{N} \), say, a thousand. Furthermore, it is not altogether justifiable to ignore (5.1) for those \( \vec{x}_j(t) \) which are widely separated from \( \vec{x}_i(t) \) in evaluating the velocity \( d\vec{x}_i/dt \). This is because \((K \cdot \Phi)(x)\) behaves like \(1/|x|\) - a Coulombic field for \(|x|\) sufficiently large. (Recall that in Section 4, we show that \(\Phi(r)\) is close to one when \(r\) is large.) Therefore it is highly a necessity to develop efficient algorithms for treating a huge number of vortex interactions, and by which the operation count could be significantly reduced.

Recently, Greengard and Rokhlin [15,16] introduced efficient algorithms for rapidly evaluating the potential field \(\phi(x)\) due to a large number \( \mathcal{N} \) of point charges \( \{q_j \delta(x - x_j)\} \) in two and three space dimensions; and by which operation counts are drastically reduced from the order \( \mathcal{N}^2 \) to the order of several \( \mathcal{N} \). Their algorithms are based on so-called multipole expansions; and can be readily incorporated into vortex algorithms to facilitate the evaluation of vortex interactions. The basic idea underlying the algorithms may be described as follows. Greengard and Rokhlin consider Poisson's equation with point charges,

\[
\Delta \phi(x) = - \sum_j q_j \delta(x - x_j), \quad x \in \mathbb{R}^n.
\]

(5.2)

The algorithms require the partitioning of a square region containing all the charges into \(4^n\) finest boxes; \(n\) is called the finest level. For a point \(x\) in a given finest box, the contribution to the potential \(\phi(x)\) from the (near-field) charges inside the box and its neighbours is
evaluated directly while the contribution from other (far-field) charges is evaluated by locally expanding finite-term multipole expansions due to these far-field charges and adding together the resultant local expansions. Specifically, let the box center be $P$ whose coordinate is $x_0$, and let $x - x_0$ have the complex coordinate $z - z_0$ in two dimensions, or the polar coordinate $(r, \theta, \varphi)$ in three dimensions. Then we have, in two dimensions,

$$
\phi(P) = \phi(P)|_{\text{near-field}} + \phi(P)|_{\text{far-field}} \\
\approx -\frac{1}{2\pi} \text{Re} \left[ \sum_{\text{near}} q_{j} \log(z - z_{j}) + \sum_{i = 0}^{p} b_{i} (z - z_{0})^{i} \right],
$$

(5.3)

where $p$ is the number of terms used in the multipole expansions. The coefficients $\{b_{i}\}$ are determined from the far-field charges. In three dimensions, the potential field is approximated by

$$
\phi(P) \approx \frac{1}{4\pi} \cdot \sum_{\text{near}} \frac{q_{j}}{|x - x_{j}|} \\
+ \frac{1}{4\pi} \cdot \sum_{i = 0}^{p} \sum_{m = -i}^{i} L_{i}^{m} \cdot Y_{i}^{m} (\theta, \varphi) \cdot r^{i},
$$

(5.4)

where $Y_{i}^{m} = Y_{i}^{m} (\theta, \varphi)$ denotes the normalized spherical harmonics. The coefficients $\{L_{i}^{m}\}$ are, again, determined from the far-field charges. The above formulas are valid for all $x$ in a given box. Therefore, in actual computation, we need only save the coefficients $\{b_{i}\}$ or $\{L_{i}^{m}\}$ for each finest box. The fast algorithms provides highly efficient procedure for obtaining the numbers $\{b_{i}\}$ and $\{L_{i}^{m}\}$ by relating local expansions at successive box levels through the use of an ‘interaction list’. (For details, we refer to the original papers.) This procedure yields an operation count of the order $pN$ provided the box level is nearly $n \approx \log_{4} N$. The formulas (5.3) and (5.4) has an error of order $1/2^{p}$. In practice, it suffices to take $p = 12$. Below we explain how the fast algorithms are employed for efficient implementation of vortex algorithms.

Consider the velocity at $(t, x)$ (cf. (2.22) or (3.23)) due to far-field vortex blobs, which is well approximated by

$$
\vec{u}_{\text{far}} (t, x) \approx \sum_{\text{far}} K(x - \vec{x}_{j} (t)) \times
$$

(5.5)

since $\Phi$ is close to one. For the sake of simplicity, in (5.5) we omit $p_{j} \omega_{j} h^{2}$ in two dimensions, and $\vec{x}_{j} (t) p_{j} h^{3}$ in three dimensions. The formula (5.5) can be obtained equivalently by
solving the Poisson equation

\[ \Delta \Psi_{\text{far}} = \sum_{\text{far}} \delta(x - \vec{x}(t)) \times , \]  

(5.6)

and taking the curl of \( \Psi_{\text{far}} \),

\[ \tilde{u}_{\text{far}}(t, x) \simeq (\nabla \times \Psi_{\text{far}})(t, x). \]  

(5.7)

The key element of the efficient implementation of vortex methods is therefore to exploit the fast algorithms for computing \( \Psi \) in (5.6), and then determine the velocity field from

\[ \tilde{u}(t, x) \simeq (\nabla \times \Psi_{\text{far}})(t, x) \\
+ \sum_{\text{near}} (K \cdot \Phi)(x - \vec{x}_j(t)) \times . \]  

(5.8)

The above formula is the one used in actual computation for approximating velocities of vortex blobs. From Tables 1 and 2, we see that in the near field, the value of \( \Phi \) changes rapidly from 0 to 1 over just several cutoff lengths, provided that the product \( \nu t \) is of the order \( \sigma^2 \). Whenever the underlying flow field encounters substantial diffusion, the near-field region expands. The help from fast algorithms is therefore diminishing. In order to avoid such situations and for additional reasons, we come up with the idea of remesh to be described precisely in the following subsection.

**Remesh procedure.** The idea is to partition the vorticity field into a new set of vortex blobs after a time of substantial diffusion. We have stressed upon the idea several times in the previous sections. In summary, the reasons of doing this are based on the following considerations:

(i) to ensure long time accuracy of vortex method,

(ii) to facilitate the implementation of fast algorithms.

Since the vortex algorithms are designed to move a whole vortex blob with a uniform velocity (induced at its center), it would be non-physical and incorrect to continue the use of the same set of vortex blobs whenever the field has encountered substantial diffusion. Furthermore, for inviscid flows, the flow field can distort much faster than viscous flows, the use of a fixed set of vortex blobs may also be physically irrelevant. Therefore if we remesh the vorticity field at an adequate time, the evolution process is refreshed on the new set of vortex blobs. Significant improvement of accuracy can therefore be achieved; and fast algorithms can still be implemented efficiently.
As to when is the proper time to remesh the vorticity field, the problem quite weighs between how viscous the involved fluid is and how fast the field distorts. Generally speaking, less viscous fluids may encounter faster distortion. Therefore, the adequate time for the remesh will not only depend on the viscosity of the fluid but also depends on the initial condition of the flow field. For very smooth flows, the typical remesh time should be several $\sigma^2/\nu$ where $\sigma$ is comparable to the mesh size $h$. The number $\sigma^2/\nu$ prevents vortex blobs from diffusing significantly into a length of several mesh sizes. Actually, for smooth slightly viscous flows, a remesh period in practice can be fairly long.

The basic formulas used in a remesh procedure are the approximate vorticity field (2.23) in two dimensions, and (3.24) in three dimensions. In the following we demonstrate the idea in two dimensions. Suppose that at $t = \bar{t}$, we remesh the vorticity field for the first time. Let $\bar{\Lambda}_h$ denote the new regular grid,

$$\bar{\Lambda}_h = \{\bar{\alpha}_j\} = \{j_1h, j_2h\},$$

and set $\tau = t - \bar{t}$, then for $t \geq \bar{t}$

$$\bar{\omega}(t, x) = \sum_{J \in \bar{\Lambda}_h} (D_r * f_\sigma)(x - \bar{x}_j(t)) \bar{p}_j \bar{\omega}_j \bar{h}^2 \quad (5.9)$$

where $\bar{x}_1(\bar{t}) = \bar{\alpha}_1$, and

$$\bar{\omega}_1 = \sum_{J \in \bar{\Lambda}_h} (D_i * f_\sigma)(\bar{\alpha}_i - \bar{x}_j(\bar{t})) \bar{p}_j \bar{\omega}_j \bar{h}^2. \quad (5.10)$$

The rest formulas in both the inviscid and viscous algorithms change correspondingly. Note especially that $D_i(x)$ in $\Phi$ should be changed to $D_r(x)$ after remesh. If we are interested in the tracking of an initial set of fluid particles, we need only to trace them in addition to the current set of vortex blobs. Formulas (5.9) and (5.10) are not the only way for remeshing the vorticity field, other interpolation techniques may also be adequate for this purpose.

6. Numerical Results and Discussions

In this section we investigate the accuracy of Algorithm $(T_\nu)$ with a remesh procedure by employing it to solve a model problem. Velocity errors are analyzed in the norm $\| \cdot \|_h$, defined in Section 4. We consider Eqs. (1.1) and (1.2) in two dimensions with the following initial vorticity field,

$$\omega_0(x) = \exp(-12|x|^2).$$
Since this initial field is radially symmetric, the solution to the problem is reduced the solution of Eq. (2.4), which is obtained by carrying out the convolution integral in (2.5),

$$\omega(t, x) = \frac{1}{\kappa_t} \exp \left( -\frac{12|x|^2}{\kappa_t} \right),$$

where

$$\kappa_t = 1 + 48\nu t.$$

The corresponding velocity is obtained by solving Eq. (2.10) and using (2.11),

$$u(t, x) = \frac{(-x_2, x_1)}{24|x|^2} \left( 1 - \exp \left( -\frac{12|x|^2}{\kappa_t} \right) \right).$$

Computed results are listed in Figs. 1a,1b,2a,2b,3a,3b and tabulated in Tables 3a,3b,3c,3d. The parameter values used in computation are (i): $\nu = 0.001, 0.0001, 0.00001, 0$, (ii): $h = 0.05, N = 1260, \sigma = 0.042, 0.05, 0.06$, while the fourth-order cutoff function $f$ from Table 1 is adopted, and the time integration is performed by Runge-Kutta methods of second order for $\Delta t = 0.08$. In the figures, we compare the accuracy of vortex algorithms with or without remesh. In Tables 3a-3d we consider the evolution of a fluid particle which is initially located at $(r, \theta) = (0.326, 0)$ for various cases of viscosity; $\beta$ denotes $\arctan(u_2/u_1)$.

We observe from these data that in order to get accurate numerical results, for a given $h$, the inviscid flow should take a larger cutoff size $\sigma$. This is certainly due to inviscid flows distorting faster than viscous flows, as explained in the previous section. Actually, we found that for larger viscosity, the cutoff size can be taken to be smaller, see for example [6]. Furthermore, the remesh procedure drastically reduces the errors by a factor from 3 to 20 (at $T=12$); however, each remesh causes an error jump due to the adoption of a new set of vortex blobs. We also observe from Table 3 that the effect of viscous diffusion is not negligible even though $\nu$ is small, say, $\nu = 0.001$, where the velocity decreases by twenty percents from the inviscid velocity at $T = 12$, and the phase angles $\theta$ and $\beta$ also lag behind by twenty degrees. From the above observations, we may conclude that (a) the effect of viscous diffusion is incorporated in an adequate way in that high order accuracy is achieved by viscous algorithms, and (b) the idea of the remesh procedure is indeed physically relevant; it causes drastic reduction of errors for both inviscid and viscous flows.

Therefore, even though the core spreading vortex method was previously shown to approximate wrong equations, the idea of restricting core spreading plus remesh proves to
be adequate and useful in approximating slightly viscous flows. Nevertheless, the present algorithms are not the only methods for approximating viscous flows. Another interesting approach is to apply the Laplace operator to the vorticity field which is represented by a sum of blob-functions, see [12], however, which does not treat the effect of viscous diffusion exactly.

In spite of the success of the restricted core spreading, we do not project that these algorithms be used for flows with large viscosity, say, $\nu > 0.01$, in which case many traditional methods may perform much better. Further success of vortex methods depends much on their capability for no-slip boundary treatment for large Reynolds number flows. Much effort has been devoted to the latter subject primarily in the content of random vortex algorithms [7,8,9]. Some recent applications along this direction can be found in [11,18,21]. On the other hand, the authors have recently extended the methodology presented in this paper to develop a reliable deterministic numerical algorithm for boundary-layer approximation. The results will be reported elsewhere. Finally we would like to remark that to the authors' opinion, though deterministic methods are suitable for the study of problems of transition and instability, random vortex algorithms would find its great value in simulating fully developed turbulent flows at large Reynolds numbers.

Acknowledgment. The first author wishes to acknowledge the support of the IMA at the University Minnesota during his stay as a Visiting Postdoctoral Member.
References

Table 1. Some useful cutoff functions in two dimensions

Second order:
\[ f(r) = \frac{1}{\pi} \exp(-r^2), \]
\[ \Phi(r) = 1 - \exp\left(-\frac{r^2}{\sigma^2 + 4\nu t}\right) (\nu \geq 0); \]

Fourth order:
\[ f(r) = \frac{1}{\pi} \left( 2 \exp(-r^2) - \frac{1}{2} \exp(-\frac{r^2}{2}) \right), \]
\[ \Phi(r) = 1 - 2 \exp\left(-\frac{r^2}{\sigma^2 + 4\nu t}\right) + \exp\left(-\frac{r^2}{2(\sigma^2 + 2\nu t)}\right) (\nu \geq 0). \]

Table 2. Some useful cutoff functions in three dimensions

Second order:
\[ f(r) = \frac{1}{\pi^{1/3}} \exp(-r^2), \]
\[ \Phi(r) = \text{erf}\left(\frac{r}{\sqrt{(\sigma^2 + 4\nu t)}}\right) - \frac{2r}{\sqrt{\pi(\sigma^2 + 4\nu t)}} \exp\left(-\frac{r^2}{\sigma^2 + 4\nu t}\right) (\nu \geq 0); \]

Fourth order:
\[ f(r) = \frac{1}{(\pi)^{1/3}} \left( 2 \exp(-r^2) - \frac{1}{2^{1/3}} \exp(-\frac{r^2}{2}) \right), \]
\[ \Phi(r) = 2 \left[ \text{erf}\left(\frac{r}{\sqrt{\sigma^2 + 4\nu t}}\right) - \frac{2r}{\sqrt{\pi(\sigma^2 + 4\nu t)}} \exp\left(-\frac{r^2}{\sigma^2 + 4\nu t}\right) \right]
- \left[ \text{erf}\left(\frac{r}{\sqrt{2\sigma^2 + 4\nu t}}\right) - \frac{2r}{\sqrt{\pi(2\sigma^2 + 4\nu t)}} \exp\left(-\frac{r^2}{2\sigma^2 + 4\nu t}\right) \right] (\nu \geq 0). \]
Figure 1a. $\sigma = 0.042$ (no remesh)

Figure 1b. $\sigma = 0.042$ (with remesh)
Figure 2a. σ = 0.05 (no remesh)

Figure 2b. σ = 0.05 (with remesh)
Notations List

\( \mathbf{x} \) : space variable
\( x_i \) : i-th space variable
\( \tilde{x}_i(t) \) : approximate position of i-th vortex blob at time \( t \)
\( \mathbf{u} \) : velocity field
\( u_i \) : i-th velocity component
\( \tilde{\mathbf{u}} \) : approximate velocity field
\( \tilde{u}_i(t) \) : approximate velocity of i-th vortex blob at time \( t \)
\( h \) : mesh size
\( f \) : cutoff function
\( D_i \) : diffusion kernel
\( K \) : velocity kernel
\( \alpha \) : initial space variable
\( \alpha_i \) : initial center of i-th vortex blob
\( \nu \) : viscosity
\( \sigma \) : cutoff size
\( \omega \) : scalar vorticity field
\( \tilde{\omega} \) : approximate scalar vorticity field
\( \varpi \) : vector vorticity field
\( \tilde{\varpi}_i(t) \) : approximate vector vorticity of i-th vortex blob at time \( t \)
\( \varpi_i(t) \) : vorticity vector associated with i-th vortex blob at time \( t \)
\( \phi \) : potential function
\( \Phi \) : function that takes care of the diffusion effect
\( \psi \) : scalar stream function
\( \Psi \) : vector stream function
<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>495</td>
<td><strong>Lucas Hsu, Niky Kamran and Peter J. Olver</strong>, Equivalence of Higher Order Lagrangians II. The Cartan Form for Particle Lagrangians</td>
</tr>
<tr>
<td>496</td>
<td><strong>D.J. Kaup and Peter J. Olver</strong>, Quantization of BiHamiltonian Systems</td>
</tr>
<tr>
<td>497</td>
<td><strong>Metin Arik, Fahirnisa Neyzi, Yavuz Nutku, Peter J. Olver and John M. Verosky</strong>, Multi-Hamiltonian Structure of the Born-Infeld Equation</td>
</tr>
<tr>
<td>498</td>
<td><strong>David H. Wagner</strong>, Detonation Waves and Deflagration Waves in the One Dimensional ZND Model for High Mach Number Combustion</td>
</tr>
<tr>
<td>499</td>
<td><strong>Jerrold R. Griggs and Daniel J. Kleitman</strong>, Minimum Cutsets for an Element of a Boolean Lattice</td>
</tr>
<tr>
<td>500</td>
<td><strong>Dieter Jungnickel</strong>, On Affine Difference Sets</td>
</tr>
<tr>
<td>501</td>
<td><strong>Pierre Leroux</strong>, Reduced Matrices and q-log Concavity Properties of q-Stirling Numbers</td>
</tr>
<tr>
<td>502</td>
<td><strong>A. Naraian and Y. Kizilyalli</strong>, The Flow of Pure Vapor Undergoing Film Condensation Between Parallel Plates</td>
</tr>
<tr>
<td>503</td>
<td><strong>Donald A. French</strong>, On the Convergence of Finite Element Approximations of a Relaxed Variational Problem</td>
</tr>
<tr>
<td>504</td>
<td><strong>Yisong Yang</strong>, Computation, Dimensionality, and Zero Dissipation Limit of the Ginzburg-Landau Wave Equation</td>
</tr>
<tr>
<td>505</td>
<td><strong>Jürgen Sprekels</strong>, One-Dimensional Thermomechanical Phase Transitions with Non-Convex Potentials of Ginzburg-Landau Type</td>
</tr>
<tr>
<td>506</td>
<td><strong>Yisong Yang</strong>, A Note On Nonabelian Vortices</td>
</tr>
<tr>
<td>507</td>
<td><strong>Yisong Yang</strong>, On the Abelian Higgs Models with Sources</td>
</tr>
<tr>
<td>508</td>
<td><strong>Chjan. C. Lim</strong>, Existence of Kam Tori in the Phase Space of Vortex Systems</td>
</tr>
<tr>
<td>509</td>
<td><strong>John Weiss</strong>, Bäcklund Transformations and the Painlevé Property</td>
</tr>
<tr>
<td>510</td>
<td><strong>Pu Fu-cho and D.H. Sattinger</strong>, The Yang-Baxter Equation for Integrable Systems</td>
</tr>
<tr>
<td>511</td>
<td><strong>E. Bruce Pitman and David G. Schaeffer</strong>, Instability and Ill-Posedness in Granular Flow</td>
</tr>
<tr>
<td>512</td>
<td><strong>Brian A. Coomes</strong>, Polynomial Flows on $C^n$</td>
</tr>
<tr>
<td>514</td>
<td><strong>Peter J. Olver</strong>, Invariant Theory, Equivalence Problems, and the Calculus of Variations</td>
</tr>
<tr>
<td>515</td>
<td><strong>Daniel D. Joseph and Thomas S. Lundgren</strong>, with an appendix by R. Jackson and D.A. Saville, Ensemble Averaged and Mixture Theory Equations</td>
</tr>
<tr>
<td>517</td>
<td><strong>Daniel D. Joseph</strong>, Generalization of the Foscolo-Gibilaro Analysis of Dynamic Waves</td>
</tr>
<tr>
<td>518</td>
<td><strong>A. Naraian and D.D. Joseph</strong>, Note on the Balance of Energy at a Phase Change Interface</td>
</tr>
<tr>
<td>519</td>
<td><strong>Daniel D. Joseph</strong>, Remarks on inertial radii, persistent normal stresses, secondary motions, and non-elastic extensional viscosities</td>
</tr>
<tr>
<td>520</td>
<td><strong>D. D. Joseph</strong>, Mathematical Problems Associated with the Elasticity of Liquids</td>
</tr>
<tr>
<td>521</td>
<td><strong>Henry C. Simpson and Scott J. Spector</strong>, Some Necessary Conditions at an Internal Boundary for Minimizers in Finite Elasticity</td>
</tr>
<tr>
<td>522</td>
<td><strong>Peter Gritzmann and Victor Klee</strong>, On the 0-1 Maximization of Positive Definite Quadratic Forms</td>
</tr>
<tr>
<td>523</td>
<td><strong>Fu-Cho Pu and D.H. Sattinger</strong>, The Yang-Baxter Equations and Differential Identities</td>
</tr>
<tr>
<td>524</td>
<td><strong>Avner Friedman and Fernando Reitich</strong>, A Hyperbolic Inverse Problem Arising in the Evolution of Combustion Aerosol</td>
</tr>
<tr>
<td>525</td>
<td><strong>E.G. Kalnins, Raphael D. Levine and Willard Miller, Jr.</strong>, Conformal Symmetries and Generalized Recurrences for Heat and Schrödinger Equations in One Spatial Dimension</td>
</tr>
<tr>
<td>526</td>
<td><strong>Wang Jinghua and Gerald Warnecke</strong>, On Entropy Consistency of Large Time Step Godunov and Glimm Schemes</td>
</tr>
<tr>
<td>527</td>
<td><strong>C. Guilloté and J.C. Saut</strong>, Existence Results for the Flow of Viscoelastic Fluids with a Differential Constitutive Law</td>
</tr>
<tr>
<td>528</td>
<td><strong>H.L. Bodlaender, P. Gritzmann, V. Klee and J. Van Leeuwen</strong>, Computational Complexity of Norm-Maximization</td>
</tr>
<tr>
<td>529</td>
<td><strong>Li Ta-tsien (Li Da-qian) and Yu Xin</strong>, Life-Span of Classical Solutions to Fully Nonlinear Wave Equations</td>
</tr>
<tr>
<td>530</td>
<td><strong>Jong-Shenq Guo</strong>, A Variational Inequality Associated with a Lubrication Problem</td>
</tr>
<tr>
<td>531</td>
<td><strong>Jong-Shenq Guo</strong>, On the Semilinear Elliptic Equation $\Delta w - \frac{1}{2} y \cdot \nabla w + \lambda w - w^{-\beta} = 0$ in $\mathbb{R}^n$</td>
</tr>
<tr>
<td>532</td>
<td><strong>Andrew E. Yagle</strong>, Inversion of the Bloch transform in magnetic resonance imaging using asymmetric two-component inverse scattering</td>
</tr>
</tbody>
</table>
Recent IMA Preprints (Continued)

<table>
<thead>
<tr>
<th>#</th>
<th>Author/s</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>533</td>
<td>Bei Hu</td>
<td>A Fiber Tapering Problem</td>
</tr>
<tr>
<td>534</td>
<td>Peter J. Olver</td>
<td>Canonical Variables for BiHamiltonian Systems</td>
</tr>
<tr>
<td>535</td>
<td>Michael Renardy</td>
<td>A Well-Posed Boundary Value Problem for Supercritical Flow of Viscoelastic Fluids of Maxwell Type</td>
</tr>
<tr>
<td>536</td>
<td>Michael Renardy</td>
<td>Ill-Posedness Resulting from Slip As a Possible Explanation of Melt Fracture</td>
</tr>
<tr>
<td>537</td>
<td>Michael Renardy</td>
<td>Compatibility Conditions at Corners Between Walls and Inflow Boundaries for Fluids of Maxwell Type</td>
</tr>
<tr>
<td>538</td>
<td>Rolf Rees</td>
<td>The Spectrum of Restricted Resolvable Designs with $r = 2$</td>
</tr>
<tr>
<td>539</td>
<td>D. Lewis and J.C. Simo</td>
<td>Nonlinear stability of rotating pseudo-rigid bodies</td>
</tr>
<tr>
<td>540</td>
<td>Robert Hardt and David Kinderlehrer</td>
<td>Variational Principles with Linear Growth</td>
</tr>
<tr>
<td>541</td>
<td>San Yih Lin and Yisong Yang</td>
<td>Computation of Superconductivity in Thin Films</td>
</tr>
<tr>
<td>542</td>
<td>A. Narain</td>
<td>Pressure Driven Flow of Pure Vapor Undergoing Laminar Film Condensation Between Parallel Plates</td>
</tr>
<tr>
<td>543</td>
<td>P.J. Vassiliou</td>
<td>On Local Equivalence for Vector Field Systems</td>
</tr>
<tr>
<td>544</td>
<td>Brian A. Coomes</td>
<td>On Conditions Sufficient for Injectivity of Maps</td>
</tr>
<tr>
<td>545</td>
<td>Yanchun Zhao</td>
<td>A Class of Global Smooth Solutions of the One Dimensional Gas Dynamics System</td>
</tr>
<tr>
<td>546</td>
<td>H. Holden, L. Holden and N.H. Risebro</td>
<td>Some Qualitative Properties of $2 \times 2$ Systems of Conservation Laws of Mixed Type</td>
</tr>
<tr>
<td>547</td>
<td>M. Slemrod</td>
<td>Dynamics of Measured Valued Solutions to a Backward-Forward Heat Equation</td>
</tr>
<tr>
<td>548</td>
<td>Avner Friedman and Jürgen Sprekels</td>
<td>Steady States of Austenitic-Martensitic-Domains in the Ginzburg-Landau Theory of Shape Memory Alloys</td>
</tr>
<tr>
<td>549</td>
<td>Avner Friedman and Bei Hu</td>
<td>Degenerate Hamilton–Jacobi–Bellman Equations in a Bounded Domain</td>
</tr>
<tr>
<td>550</td>
<td>E.G. Kalnins, Willard Miller, Jr., and M.V. Tratnik</td>
<td>Families of Orthogonal and Biorthogonal Polynomials on the N-Sphere</td>
</tr>
<tr>
<td>551</td>
<td>Heinrich Freistühler</td>
<td>On Compact Linear Degeneracy</td>
</tr>
<tr>
<td>552</td>
<td>Matthew Witten</td>
<td>Quantifying the Concepts of Rate and Acceleration/Deceleration of Aging</td>
</tr>
<tr>
<td>553</td>
<td>J.P. Albert and J.L. Bona</td>
<td>Total Positivity and the Stability of Internal Waves in Stratified Fluids of Finite Depth</td>
</tr>
<tr>
<td>554</td>
<td>Brian Coomes and Victor Zurkowski</td>
<td>Linearization of Polynomial Flows and Spectra of Derivations</td>
</tr>
<tr>
<td>555</td>
<td>Yuriko Renardy</td>
<td>A Couette-Poiseuille Flow of Two Fluids in a Channel</td>
</tr>
<tr>
<td>556</td>
<td>Michael Renardy</td>
<td>Short wave instabilities resulting from memory slip</td>
</tr>
<tr>
<td>557</td>
<td>Daniel D. Joseph and Michael Renardy</td>
<td>Stokes’ first problem for linear viscoelastic fluids with finite memory</td>
</tr>
<tr>
<td>558</td>
<td>Xiaxi Ding</td>
<td>Superlinear Conservation Law with Viscosity</td>
</tr>
<tr>
<td>559</td>
<td>J.L. Ericksen</td>
<td>Liquid Crystals with Variable Degree of Orientation</td>
</tr>
<tr>
<td>560</td>
<td>P. Robert Ore, Jr. and Xinfu Chen</td>
<td>Electro-Optic Modulation in an Arbitrary Cross-Section Waveguide</td>
</tr>
<tr>
<td>561</td>
<td>M.V. Tratnik</td>
<td>Multivariable biorthogonal continuous-discrete Wilson and Racah polynomials</td>
</tr>
<tr>
<td>562</td>
<td>Yisong Yang</td>
<td>Existence of Solutions for a Generalized Yang-Mills Theory</td>
</tr>
<tr>
<td>563</td>
<td>Peter Gritzmann, Laurent Habsieger and Victor Klee</td>
<td>Good and Bad Radii of Convex Polygons</td>
</tr>
<tr>
<td>564</td>
<td>Martin Golubitsky, Martin Krupa and Chian. C. Lim</td>
<td>Time-Reversibility and Particle Sedimentation</td>
</tr>
<tr>
<td>565</td>
<td>G. Yin</td>
<td>Recent Progress in Parallel Stochastic Approximations</td>
</tr>
<tr>
<td>566</td>
<td>G. Yin</td>
<td>On II-Valued SA: Finite Dimensional Approximations</td>
</tr>
<tr>
<td>567</td>
<td>Chien-Cheng Chang</td>
<td>Accurate Evaluation of the Effect of Diffusion and Conductivity in Certain Equations</td>
</tr>
<tr>
<td>568</td>
<td>Chien-Cheng Chang and Ruey-Ling Chern</td>
<td>The Effect of Viscous Diffusion in Discrete Vortex Dynamics for Slightly Viscous Flows</td>
</tr>
<tr>
<td>569</td>
<td>Li Ta-Tsien (Li Da-qian) and Zhao Yan-Chun</td>
<td>Global Existence of Classical Solutions to the Typical Free Boundary Problem for General Quasilinear Hyperbolic Systems and its Applications</td>
</tr>
<tr>
<td>570</td>
<td>Thierry Cazenave and Fred B. Weissler</td>
<td>The Structure of Solutions to the Pseudo-Conformally Invariant Nonlinear Schrödinger Equation</td>
</tr>
<tr>
<td>571</td>
<td>Marshall Slemrod and Athanasios E. Tzavaras</td>
<td>A Limiting Viscosity Approach for the Riemann Problem in Isentropic Gas Dynamics</td>
</tr>
<tr>
<td>573</td>
<td>P.J. Vassiliou</td>
<td>On the Geometry of Semi-Linear Hyperbolic Partial Differential Equations in the Plane Integrable by the Method of Darboux</td>
</tr>
<tr>
<td>574</td>
<td>Jerome V. Moloney and Alan C. Newell</td>
<td>Nonlinear Optics</td>
</tr>
<tr>
<td>575</td>
<td>Keti Tenenblat</td>
<td>A Note on Solutions for the Intrinsic Generalized Wave and Sine-Gordon Equations</td>
</tr>
<tr>
<td>576</td>
<td>P. Szmolyan</td>
<td>Heteroclinic Orbits in Singularly Perturbed Differential Equations</td>
</tr>
<tr>
<td>577</td>
<td>Wenxiong Liu</td>
<td>A Parabolic System Arising In Film Development</td>
</tr>
</tbody>
</table>