HIGH ORDER VORTEX METHODS WITH DEFORMING ELLIPTICAL GAUSSIAN BLOBS 1: DERIVATION AND VALIDATION *

LOUIS F. ROSSI†

Key words. vortex methods, vorticity dynamics, computational fluid dynamics, convergence theory

AMS subject classifications. 35Q30, 41A25, 41A30, 65D99, 65M12, 65M50, 65M60, 76D05

Abstract. This manuscript introduces a new vortex method based on elliptical Gaussian basis functions. Each basis function translates, nutates, elongates and spreads through the action of the local flow field and fluid viscosity. By allowing elements to deform, the method captures the effects of local flow deviations with a fourth order spatial accuracy. This method uses a fourth order asymptotic approximation to the Biot-Savart integrals for elliptical Gaussian vorticity distributions to determine velocity and velocity derivatives. A robust adaptive refinement procedure reconfigures elements that spread beyond the specified resolution. The high order convergence rate is verified by comparing calculations with the vortex method to exact solutions in a variety of controlled experiments.

1. Introduction. This paper discusses a new, high spatial order vortex method that approximates the vorticity field of a fluid flow with elliptical Gaussian basis functions that translate, spread, elongate and rotate with the flow. The primary justification for studying the properties of this method is to perform practical calculations with a small number of computational elements, but also the method possesses interesting mathematical properties that are interesting in and of themselves.

Vortex methods, arguably more so than any other scheme used to compute fluid flows, closely resemble a simplified physical model because the evolution of each computational element approximates the evolution of a patch of vorticity. For certain flows such as the Kida vortex in a linear flow, a "vortex method" consisting of a single elliptical patch would represent an exact solution to the Euler equations [7]. In the context of this article, no such model exists for an elliptical Gaussian region of vorticity because this region is not necessarily self-similar under the action of the Navier-Stokes equations.

Vortex methods approximate the vorticity field as linear combination of either delta functions (point vortices) or localized basis functions (vortex blobs). The latter can be viewed as a regularization of the Biot-Savart kernel. The point vortex method dates back to Rosenhead's 1932 calculation of vortex sheet instabilities [21]. In a point vortex calculation, each point vortex moves with the flow velocity measured at the point vortices' position, and the greatest difficulty in calculations using point vortices is the singularity in the induced velocity field. As one increases the spatial resolution and the distance between computational elements decreases, the velocity field increases asymptotically requiring smaller time steps. However, despite the stiffness in point vortex dynamics, these methods are convergent in the sense that trajectories of the computational elements will approach exact trajectories as one uses more and more point vortices to approximate the field as shown by Goodman, Hou and Lowengrub [3]. However, there is no clear advantage to such a method over regularized

*Portions of this work was supported by NSF grant DMS-9407660; other portions supported by NSF grant DMS-9971800.

 Department of Mathematical Sciences, University of Delaware. Newark, DE 19711, USA (rossi@math.udel.edu)

methods which avoid the stiffness issue entirely, especially if the goal is to examine solutions to the Navier-Stokes equations where viscous diffusion must be incorporated somehow.

Regularizing the Biot-Savart velocity calculation is equivalent to using localized distributions of vorticity rather than point vortices as computational elements, and understanding how the basis functions evolve is the key to understanding the accuracy of the vortex method. The computational elements move with the flow velocity at the centroid determined from a Biot-Savart integral. The shape of the blob is justified either mathematically or physically as a regularization of the delta function. For instance, Gaussian basis functions are a natural viscous regularization of a delta function source of vorticity. Usually, though not always, the basis functions are radially symmetric, and the Biot-Savart integral has an analytic solution. References on a wide variety of vortex method implementations can be found in [2, 12, 25]. In studying the convergence properties of vortex methods, Beale and Majda proved that certain radially symmetric functions would yield higher spatial accuracy based on moment conditions [1, 12]. These moment conditions arise when integrating the basis function against a Taylor expansion of the vorticity field. Typically, this inviscid analysis decomposes the approximation into two parts, smoothing error and discretization error. The former is caused by regularizing the Biot-Savart kernel, and the latter is caused by interpolating the vorticity field with a finite array of elements separated by some characteristic distance and is related to the number of continuous derivatives in the basis function. Winkelsmans and Leonard provide good survey of various regularizations (basis function shapes) and their accuracies [31]. For instance, a Gaussian basis function has second order smoothing error and infinite order discretization error. By selecting an accurate regularization, one can boost the spatial accuracy of the inviscid part of the vorticity calculation. In another approach, Lowengrub and Shelley have boosted the order of Lagrangian schemes by coupling point vortex dynamics to an underlying curvilinear coordinates system that does not necessarily move with the fluid velocity as Lagrangian methods do [14]. Later, with Merriman, they expanded this work by incorporating high-order corrections on a rectangular grid to achieve fourth-order spatial accuracy [15]. A third approach is to have the dynamics of the basis function coupled to the velocity field and its derivatives. This approach permits more adaptivity in the vortex method toward whatever purpose concerns the investigator. Furthermore, it is not necessary to restrict one's analysis to the Euler equations and then treat viscosity as a separate issue. The resulting method would require the computational elements to have additional degrees of freedom to capture the dynamics of the velocity field expansion. These additional degrees of freedom correspond to a deforming basis function or "blob."

Many investigators have explored both non-axisymmetric and deforming basis functions in the hopes that they would adaptively resolve vorticity fields and passive scalar fields. Meiburg developed a scheme in which radially symmetric blobs representing sections of a vortex sheet would expand or contract based on normal flow deviations, but this scheme violates incompressibility and is specialized for vortex sheets [18]. Marshall and Grant have used highly anisotropic elements to satisfy the no-slip, no normal flow boundary conditions efficiently. Teng originally used rigid elliptical patches to resolve boundary layers more efficiently and later used deforming elliptical patches to simulate the evolution of vortex sheets [27, 28, 29]. This work was not coupled to any adaptive refinement techniques though Teng does refer to the necessity of some sort of refinement to make these methods work, and he establishes

a theoretical $O(l^2 \log l)$ rate of convergence for his method. Kida et. al. have used simple core spreading and a second order core spreading method. The secord order method is similar to that of Lu and Rossi based on an integral correction and continual regridding of the computational elements [16]. This group has shown that core spreading without refinement or corrections is valid for small times while the corrected method is valid for arbitrary finite times [10, 8, 9]. Independently, Rossi proposed a corrected core spreading vortex method (CCSVM) which refines elements that have grown beyond a specified core size [22]. This maintains spatial resolution for all finite time at a cost of introducing more computational elements. Later, Shiels studied the refinement process in greater detail [26]. To simulate the convection *and diffusion* of passive scalar quantities, Leonard used deforming elliptical Gaussian basis functions, noting that they remain self-similar under the linearized convection-diffusion equations [13]. Up to this point, such approaches have not been considered for vortex methods because there is no known Biot-Savart integral for elliptical Gaussians that can be expressed in terms of elementary functions. Ojima & Kamemoto propose a scheme using deforming elements that stretch with local flow deviations, but the resulting element is replaced with an isotropic element of equal volume at the end of each timestep, thus avoiding having to calculate the Biot-Savart integral for anisotropic elements [19].

This paper describes the essentials and some of the practicalities of the Elliptical Corrected Core Spreading Vortex Method (ECCSVM), but some issues are left to forthcoming manuscripts. This paper is self-contained in the sense that the method is completely defined and analyzed. However, several issues that enhance its performance considerably are beyond the scope of this work. Specifically, the error bounds and details of the merging algorithm are left to a future article and do not add anything to the understanding of ECCSVM. However, merging techniques do enhance the efficiency of ECCSVM considerably. Also, fast summation methods make this method substantially more practical as problem sizes grow large, but are also beyond the scope of this paper. §2 is a broad overview of the numerical method. I derive the ODEs governing the motion and evolution of elliptical basis functions in §3, and discuss some of their properties. I also describe the behavior of this method relative to integral invariants of the Navier-Stokes equations. I analyze the residual of linearized vorticity dynamic operator applied to this method and determine a $O(l^4)$ spatial accuracy. In this section, I also calculate a fourth order asymptotic expansion of the velocity and velocity deviations fields induced by an elliptical Gaussian vorticity field because there is no known expression for these Biot-Savart integrals as simple combinations of elementary functions. In §4, I describe a spatial refinement procedure for replacing a wide computational element with a configuration of thinner elements while introducing a small controllable error. In §5, I perform some sample calculations and verify that the anticipated rate of convergence is achieved. In §6, I discuss where one can obtain the source code for ECCSVM and how one can reproduce all the results discussed all the results discussed in this manuscript.

2. Natural Adaptivity and the Need for Refinement and Merging. Though the principle advantage of Lagrangian methods is their natural adaptivity, Lagrangian schemes also have some significant shortcoming, particularly growth in the number of computational elements. When simulating flows near boundaries, vorticity is created through the no-slip boundary condition and thus new computational elements must be introduced into the flow. Though redistribution techniques do not appear to suffer from this shortcoming, they rely upon computational elements al-

ready being present near solid boundaries, sometime in the form of "ghost particles," to exchange vorticity with the wall. Even in flows without boundaries, viscous diffusion will transport vorticity across streamlines requiring some technique of generating new elements in previously irrotational regions of the flow if one wishes to maintain spatial accuracy. With core spreading methods, this occurs when the width of a given blob grows with time. To maintain spatial accuracy however, one must replace wide elements with thinner elements periodically, and so the problem size grows.

Unfortunately, not all computational elements in a vortex simulation are necessary. For instance, if several elements of the same shape overlap exactly, they can be replaced by a single element without any change in spatial accuracy. Likewise, a single element can replace overlapping dissimilar elements at the cost of introducing small errors into a simulation. A merging algorithm based on this philosophy has been analyzed for Gaussian basis functions and has played an important role in extended calculations [23, 24]. Thus, a core spreading algorithm will have the following general structure.

1. Initialize all vortex elements.
2. **The computational elements move and evolve.** Solve the system of ODEs governing the evolution of the computational elements.
3. **New elements are generated as needed.** If any elements grow wider than a specified tolerance, replace them with a configuration of thinner elements.
4. **Redundant elements are merged with one another, if possible.** If a collection of elements can be replaced by a single element while introducing an error beneath a specified threshold, replace the collection with a single element.
5. Return to step 2.

This algorithm can be generalized to other vortex methods besides ECCSVM. For instance, vorticity redistribution strategies must introduce new elements to diffuse vorticity away from compact regions. In any method with solid boundaries, vorticity must be introduced to satisfy the no-slip boundary condition. Naturally, a merging algorithm would benefit any method.

The issue of problem size control is different from gridded, spectral or finite element methods. For the latter methods, the problem size is specified and the resolution together with the qualities of the exact solution determine the quality of the approximation. For instance, errors enter into any calculation if the computational domain lacks proper coverage for a given problem. On the other hand, core spreading Lagrangian methods will always maintain adequate coverage of the physical domain, but the specified resolution may require the problem size to grow. For adaptive schemes, the quality of the solution is specified by the resolution, and the problem size grows to accommodate the specified tolerances.

3. The Vortex Method. This section describes the specifics of ECCSVM. Several aspects make it more complex than most Lagrangian schemes. For one, it arises from a higher order approximation of the Navier-Stokes equations. Also, basis functions deform as well as move with the flow. The differential equations governing this deformation, when expressed in terms of geometric variables, have singularities that must be considered. Also, in order to establish convergence properties, one must bound certain basis function parameters. With these results, one can then analyze the spatial accuracy of the scheme. Finally, one must find an accurate and computationally feasible means of calculating flow velocities and derivatives induced by an elliptical Gaussian distribution of vorticity.

3.1. The Continuous Formulation of Convergence. The governing equations are the two-dimensional, incompressible Navier-Stokes equations which can be written in terms of vorticity, $\omega(x, y, t)$:

$$\begin{aligned} \partial_t \omega + (\vec{u} \cdot \nabla)\omega &= \nu \nabla^2 \omega \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \tag{3.1}$$

where $\vec{u}(x, y, t)$ is the fluid velocity and $\omega = (\nabla \times \vec{u}) \cdot \hat{z}$. Since the flow is two-dimensional, the vorticity is directed exclusively in the vertical direction and is treated as a scalar quantity.

In previous work on core spreading vortex methods, an alternative formulation of convergence is proposed wherein the spatial accuracy is measured in terms of the ability of the semi-discrete vortex method to approximate solutions of the Navier-Stokes equations [12, 22]. To quantify the spatial accuracy for any Lagrangian method for solving (3.1), Leonard proposed examining the continuous residual of the approximate scheme. This residual can be defined in terms of an operator

$$Rf = \partial_t f + \vec{u} \cdot \nabla f - \nu \nabla^2 f \tag{3.2}$$

where $f(x, y, t)$ is any sufficiently differentiable function and $\nabla \cdot \vec{u} = 0$. For the fully nonlinear Navier-Stokes operator. $f = \nabla \times \vec{u}$. To estimate the spatial accuracy of the high order method, we consider the linearized operator where $\vec{u}$ is a known function, not determined by f. Thus, the exact velocity field $\vec{u}$ and its derivatives are known, bounded quantities in this analysis. Within this framework, f is any function including an exact solution or an approximate solution obtained using a numerical method, and we refer to Rf as the *residual* or error associated with f at each point in space and time. If ω designates an exact solution to (3.1), then

$$R\omega = 0. \tag{3.3}$$

If we let $\widehat{\omega}(x, y, t)$ designate an approximation using some numerical scheme, in general we do not expect the residual, $R\widehat{\omega}$, to be zero.

For a core spreading vortex method where the vorticity field is discretized into blobs that move and spread as usual

$$\begin{aligned} \widehat{\omega} &= \sum_{i=0}^{N} f(\gamma_i, \vec{x} - \vec{x}_i, \sigma_i) \\ f(\gamma, \vec{x}, \sigma) &= \frac{\gamma}{4\pi\sigma^2} \exp\left(\frac{|\vec{x}|^2}{4\sigma^2}\right), \end{aligned} \tag{3.4}$$

one finds that

$$R\widehat{\omega} = \sum_{i=0}^{N} \nabla_{\vec{x}} f(\gamma_i, \vec{x} - \vec{x}_i, \sigma_i) \cdot [\vec{u}(\vec{x}_i) - \vec{u}(\vec{x})], \tag{3.5}$$

suggesting that this residual quantity is bounded by differences between the exact and approximate velocity across the support of each basis function [22]. That is to say, each basis function satisfies the local equation

$$\partial_t f + \vec{u}(\vec{x}_i) \cdot \nabla f - \nu \nabla^2 f = 0. \tag{3.6}$$

If a basis function, g, satisfied the local equation

$$\partial_t g + [\vec{u}(\vec{x}_i) + D\vec{u}(\vec{x}_i)(\vec{x} - \vec{x}_i)] \cdot \nabla g - \nu \nabla^2 g = 0, \tag{3.7}$$

where $D\vec{u}$ is the matrix of partial derivatives of $\vec{u}$, the residual for $\widehat{\omega}$ would be

$$R\widehat{\omega} = \sum_{i=0}^{N} \nabla_{\vec{x}} g \cdot [\vec{u}(\vec{x}_i) + D\vec{u}(\vec{x}_i)(\vec{x} - \vec{x}_i) - \vec{u}(\vec{x})]. \tag{3.8}$$

Once can see that (3.7) is a closer approximation to (3.2) and so (3.8) should be of higher order. A Gaussian will not satisfy (3.7), but an elliptical Gaussian will, and this is the basis of the high spatial order vortex method.

3.2. The Evolution of Elliptical Gaussians. Elliptical Gaussians have five internal degrees of freedom: position, $\vec{x}_i(t)$, orientation, $\theta_i(t)$, major axis, $a_i(t)$, and core width, $\sigma_i(t)$. To approximate a solution to the Navier-Stokes equations, these basis functions translate, rotate, elongate and spread under the convective action of the linearized flow field and the diffusive action of viscosity. A single basis function has the form

$$g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i) = \\ \frac{\gamma_i}{4\pi\sigma_i^2} \exp\left\{ -\frac{[c_i(x - x_i) + s_i(y - y_i)]^2/a_i^2 + [-s_i(x - x_i) + c_i(y - y_i)]^2 a_i^2}{4\sigma_i^2} \right\} \tag{3.9}$$

where $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$. This formulation where the product of the semi-major and semi-minor axes is unity isolates the core size parameter, σ_i, and the aspect ratio, a_i^2. Implementing geometric variables rather than a general quadratic form as Leonard and Teng have done permits one to more easily control the relevant convergence parameters, but the distinction is a minor one for all intents and purposes. Of course, the entire vorticity field is approximated as a linear combination of these basis functions.

$$\widehat{\omega}(\vec{x}, t) = \sum_{i=0}^{N} g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i) \tag{3.10}$$

With the exception of γ_i, all indexed parameters evolve with time.

To derive the evolution equations for the basis function parameters, we apply (3.8),

$$\partial_t g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i) + [\vec{u}(\vec{x}_i) + D\vec{u}(\vec{x}_i)(\vec{x} - \vec{x}_i)] \cdot \nabla_{\vec{x}} g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i) = \\ \nu \nabla^2 g(\vec{x}; \gamma_i, \vec{x}_i, \sigma_i, a_i, \theta_i). \tag{3.11}$$

where $\vec{u}(\vec{x})$ and $D\vec{u}(\vec{x})$ arise from Biot-Savart integrations on the Gaussian basis functions. Several expansions and manipulations, explained more completely in Appendix A, yield the following discrete dynamical system describing the motion and deforma-

tion of the computational elements:

$$\begin{aligned}
\frac{d}{dt}\vec{x}_i &= \begin{bmatrix} u_i \\ v_i \end{bmatrix} \\
\frac{d}{dt}(\sigma_i^2) &= \frac{\nu}{2}(a_i^2 + a_i^{-2}) \\
\frac{d}{dt}(a_i^2) &= 2[d_{11,i}(c_i^2 - s_i^2) + (d_{12,i} + d_{21,i})c_i s_i]a_i^2 + \frac{\nu}{2\sigma_i^2}(1 - a_i^4) \\
\frac{d}{dt}\theta_i &= \frac{d_{21,i} - d_{12,i}}{2} + \left[\frac{d_{21,i} + d_{12,i}}{2}(s_i^2 - c_i^2) + 2d_{11,i}s_i c_i\right]\frac{(a_i^{-2} + a_i^2)}{(a_i^{-2} - a_i^2)}
\end{aligned} \tag{3.12}$$

where $\vec{u}(\vec{x}_i) = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$ and $D\vec{u}(\vec{x}_i)$ has elements $d_{11,i} = \frac{du}{dx}(\vec{x}_i)$, $d_{12,i} = \frac{du}{dy}(\vec{x}_i)$, etc.

From system (3.12), one observes certain properties in these elliptical Gaussians. The first equation is no surprise. From the second equation, one can see that elongation of the basis functions augments spreading as one might expect. As with CCSVM, the width, σ_i, must be numerically controlled by adaptive refinement. This procedure will be described in more detail in §4, but here we will use the fact that the refinement procedure exists and guarantees that $\sigma_i < l$ where l is a controllable numerical parameter associated with the accuracy of the method.

In the third equation of (3.12), one can see that in the absence of viscosity, basis functions can elongate exponentially. However, the growth of the aspect ratio is bounded above by

$$a_i^2 \leq e^{2\lambda_M t} \tag{3.13}$$

where λ_M is the largest positive value of λ_i over a relevant spatial or temporal domain and

$$\lambda_i = d_{11,i}(c_i^2 - s_i^2) + (d_{12,i} + d_{21,i})c_i s_i. \tag{3.14}$$

For this linearized analysis, we assume $\vec{u}$ and its derivatives are known and bounded, so one can be certain that λ_M exists. While this paper focuses on $\nu \neq 0$, it is important to note that unlimited growth in the aspect ratio as $t \to \infty$ is catastrophic for both the spatial accuracy as discussed studied later in this section and the asymptotics for the Biot-Savart integral in §3.5.

To analyze this method, the typical domain is the trajectory of a computational element through the flow, or perhaps the entire flow over all space and time. The bound, λ_M, is valid for all indices, i and all time and assumes unit initial aspect ratios. If $\nu \neq 0$, viscosity naturally regulates the elongation, and there is an upper bound, $a_i^2 \leq a_M^2$:

$$a_M^2 = \frac{2\lambda_M l^2}{\nu} + \sqrt{\frac{4\lambda_M^2 l^4}{\nu^2} + 1} = R_{\mathcal{E}} + \sqrt{R_{\mathcal{E}}^2 + 1}. \tag{3.15}$$

Thus, the maximum elongation of each basis function is controlled by the dimensionless quantity

$$R_{\mathcal{E}} = \frac{2\lambda_M l^2}{\nu} \tag{3.16}$$

which from this point on shall be referred to as the **basis function Reynolds number** somewhat related to the grid Reynolds number used to analyze finite difference schemes.

Herein lies a hidden advantage of corrected core spreading methods. We shall see later that the spatial accuracy of this method is $O(l^2(a^2 - a^{-1}))$. Thus, an investigator might be tempted to use two numerical parameters to control l and $a^2 - a^{-1}$, respectively. In fact, we have just see that this is not necessary. For small l (and therefore small basis function Reynolds number), we can expand (3.15) as

$$a_M^2 \approx 1 + R_{\mathcal{E}} + \frac{1}{2}R_{\mathcal{E}}^2 + \ldots = 1 + O(l^2) \tag{3.17}$$

Thus, $a^2 - a^{-2} = O(l^2)$, and l is the only relevant numerical parameter for this scheme, and no additional adaptive refinement is required to force the method to converge.

In the fourth equation, we see that rotation is induced through two distinct mechanisms. The first term on the right hand side corresponds to revolution due to pure rotation in $D\vec{u}$. The second term corresponds to rotation induced by strain that is not aligned with the major or minor axis. This latter term can be a source of stiffness in the system if a_i is close to 1. If a_1 is not close to unity, the evolution equations can be solved directly.

Since one expects some aspect ratios to remain close to 1 in typical flow regimes, it is crucial that one deal with this stiffness when solving the evolution equations (3.12) for the vorticity field. When a_i is close to unity, the orientation of the element rapidly moves toward an equilibrium where the two terms in the evolution equation balance one another. Dropping indices from this point onward, we can calculate these equilibria by setting $\frac{d}{dt}\theta = 0$ in (3.12).

$$\frac{d_{21} - d_{12}}{2}(a^{-2} - a^2)(c^2 + s^2) + \left[\frac{d_{21} + d_{12}}{2}(s^2 - c^2) + 2d_{11}sc\right](a^{-2} + a^2) = 0 \tag{3.18}$$

Collecting like terms in c and s and dividing by c^2, one arrives as a quadratic equation for $\tan\theta$. Solving this equation, one finds that the four equilibria (two from the quadratic plus two from the twofold symmetry of the basis function) are described by

$$\begin{aligned} \tan(\theta^{\pm}) &= -k \pm \sqrt{k^2 + \frac{a^{-2}d_{12} + a^2 d_{21}}{a^2 d_{12} + a^{-2}d_{21}}}, \\ k &= \frac{d_{11}(a^2 + a^{-2})}{a^2 d_{12} + a^{-2}d_{21}} \end{aligned} \tag{3.19}$$

Of course, one root corresponds to a stable equilibrium and the other corresponds to an unstable equilibrium. If a is close to unity and the problem is very stiff, one can effectively and accurately integrate (3.12) by assuming that the orientation of the computational element is evolving through stable, local equilibrium so that $\theta \equiv \theta(D\vec{u}, a)$ found by solving (3.19). Therefore, one integrates only a and σ in time.

The analysis in the section can be summarized as follows:

- In a linearized regime where the velocity field is known, the basis function Reynolds number (3.16) remains bounded, independent of the dynamics of the basis function.
- If $\nu \neq 0$, the aspect ratio of every blob is bounded by (3.15).
- One can numerically integrate (3.12) directly except when a_i passes through unity. In this case, one can integrate the system by enslaving the orientation variable to the aspect ratio and width, and proceeding directly.

3.3. The High Spatial Order Residual. As noted earlier, one can examine the accuracy of a vortex method by considering the residual of (3.2). This section is not intended to be a full convergence proof. A full convergence proof would follow the same form as that which was used for CCSVM (described in [22]), but the essential changes which examine the deformation terms are contained here. We linearize the full vorticity dynamics operator, R, and suppose that the velocity field and the deviation tensor is specified. Though the full nonlinear problem is more challenging, the rate of convergence is the key issue, and the linear problem establishes the rate of convergence as $l \to 0$, because the computed velocity field converges to the true velocity field in this same limit.

A new issue that is unique to using elliptical Gaussians is that the computed velocity field is, in fact, an approximation of the Biot-Savart integral for the approximate vorticity field, $\widehat{\omega}$. In other words, the velocity field and velocity field deviation calculation is an approximation based on an approximation. As will be shown in §3.5, the approximation to the Biot-Savart integral is fourth order in $1 - a_i$. Since $1-a_i^2 = O(l^2)$ and $1-a_i^2 = (1-a_i)(1+a_i)$, the approximation to the velocity field and its derivatives converges with a $O(l^8)$ which far exceeds the overall spatial accuracy of the method.

The total residual of a computational element under the dynamics of (3.12) is

$$\begin{aligned}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R\left[g(\vec{x})\right] d\vec{x} = \int\int \nabla g \cdot [&-\frac{1}{2}D^2\vec{u}(\vec{x}_i)(\vec{x}-\vec{x}_i,\vec{x}-\vec{x}_i)\\ &-\frac{1}{3!}D^3\vec{u}(\vec{x}_i)(\vec{x}-\vec{x}_i,\vec{x}-\vec{x}_i,\vec{x}-\vec{x}_i)\\ &-\frac{1}{4!}D^4\vec{u}(\vec{x}_i)(\vec{x}-\vec{x}_i,\ldots,\vec{x}-\vec{x}_i)\\ &-\frac{1}{5!}D^5\vec{u}(\vec{\xi}_i)(\vec{x}-\vec{x}_i,\ldots,\vec{x}-\vec{x}_i)]d\vec{x},\end{aligned} \tag{3.20}$$

where $D^2\vec{u}(\vec{x}_i)$ (the "Hessian") is a second order tensor or bilinear operator of velocity derivatives evaluated at the basis function center, $\vec{x}_i$. Similarly, $D^3\vec{u}(\vec{x}_i)$ is a trilinear operator of velocity derivatives. For the last term in the expansion, $\vec{\xi}_i$ is a point on the line segment joining $\vec{x}$ and $\vec{0}$. From this point on, the tensors (including their constituents) are evaluated at $\vec{x}_i$, and references to $\vec{x}_i$ will be excluded in the interest of brevity. Without loss of generality, let us assume that the basis function (3.9) is located at the origin and oriented along the coordinate axes so that $\vec{x}_i = \vec{0}$, $c_i = 1$ and $s_i = 0$. Then,

$$\nabla g = \frac{\gamma_i}{8\pi\sigma^4}\exp\left[-\frac{\left(x^{(1)}\right)^2}{4\sigma_i^2 a_i^2} - \frac{\left(x^{(2)}\right)^2 a_i^2}{4\sigma_i^2}\right]\begin{bmatrix}\left(x^{(1)}\right)a_i^{-2}\\ \left(x^{(2)}\right)a_i^2\end{bmatrix}. \tag{3.21}$$

Here, it is convenient to use index notation so that $x^{(1)} = x$ and $x^{(2)} = y$ because we will have many repeated indices when analyzing the $D^3\vec{u}$ term of (3.20).

In the first combination of terms in (3.20), one sees that

$$\nabla g \cdot D^2\vec{u}(\vec{0})(\vec{x},\vec{x}) = \frac{\gamma_i}{8\pi\sigma_i^4}\exp\left[-\frac{\left(x^{(1)}\right)^2}{4\sigma_i^2 a_i^2} - \frac{\left(x^{(2)}\right)^2 a_i^2}{4\sigma_i^2}\right]\frac{\partial^2 u^j}{\partial_{x^k}\partial_{x^m}}x^j x^k x^m, \tag{3.22}$$

where one sums over repeated indices. Since the partial derivatives composing the Taylor coefficients are evaluated at the center of the basis function and therefore

remain fixed, any combination of j, k, m will involve an odd moment on the elliptical Gaussian. Therefore,

$$\int\int \nabla g \cdot D^2\vec{u}(\vec{0})(\vec{x},\vec{x})d\vec{x} = 0. \tag{3.23}$$

On the surface, it would appear that the second combination of terms has even moments and so will determine the rate of convergence for the whole method. I save this portion for last because it is contains a pleasant surprise in this respect. Similar to the first combination, the third combination in (3.20),

$$\iint \nabla g \cdot D^4\vec{u}(\vec{0})(\vec{x},\vec{x},\vec{x},\vec{x})d\vec{x} = 0 \tag{3.24}$$

for the same reason. The last term actually determines one piece of the convergence rate

$$\iint \nabla g \cdot D^5\vec{u}(\xi_i)(\vec{x},\ldots,\vec{x})d\vec{x} = O(l^4) \tag{3.25}$$

because $\sigma_i \leq l$.

Returning to the second combination, even moments are possible:

$$\begin{aligned}
\iint \nabla g \cdot D^3\vec{u}(\vec{0})(\vec{x},\vec{x},\vec{x})d\vec{x} &= \iint \frac{\gamma_i}{8\pi\sigma^4}\exp(\ldots)\frac{\partial^3 u^j}{\partial_{x^k}\partial_{x^m}\partial_{x^n}}x^j x^k x^m x^n. \\
&= \iint \frac{\gamma_i}{8\pi\sigma^4}\exp(\ldots)\left[\frac{\partial^3 u^{(1)}}{\partial_{\left(x^{(1)}\right)^3}}\left(x^{(1)}\right)^4 + \right. \\
&\quad 3\frac{\partial^3 u^{(1)}}{\partial_{x^{(1)}}\partial_{\left(x^{(2)}\right)^2}}\left(x^{(1)}\right)^2\left(x^{(2)}\right)^2 + \\
&\quad 3\frac{\partial^3 u^{(2)}}{\partial_{\left(x^{(1)}\right)^2}\partial_{x^{(2)}}}\left(x^{(1)}\right)^2\left(x^{(2)}\right)^2 + \\
&\quad \left.\frac{\partial^3 u^{(2)}}{\partial_{\left(x^{(2)}\right)^3}}\left(x^{(2)}\right)^4\right] d\vec{x}.
\end{aligned} \tag{3.26}$$

Applying incompressibility to the two middle expressions in the summation yields

$$\begin{aligned}
\iint \nabla g \cdot D^3\vec{u}(\vec{0})(\vec{x},\vec{x},\vec{x})d\vec{x} &= \iint \frac{\gamma_i}{8\pi\sigma^4}\frac{\partial^3 u^j}{\partial_{x^k}\partial_{x^m}\partial_{x^n}}\exp(\ldots)x^j x^k x^m x^n d\vec{x} \\
&= \iint \frac{\gamma_i}{8\pi\sigma^4}\exp(\ldots)\times \\
&\quad \left\{\frac{\partial^3 u^{(1)}}{\partial_{\left(x^{(1)}\right)^3}}\left[\left(x^{(1)}\right)^4 - 3\left(x^{(1)}\right)^2\left(x^{(2)}\right)^2\right] + \right. \\
&\quad \left.\frac{\partial^3 u^{(2)}}{\partial_{\left(x^{(2)}\right)^3}}\left[\left(x^{(2)}\right)^4 - 3\left(x^{(1)}\right)^2\left(x^{(2)}\right)^2\right]\right\} d\vec{x}.
\end{aligned} \tag{3.27}$$

Evaluating the integral, one finds

$$\iint \nabla g \cdot D^3\vec{u}(\vec{0})(\vec{x},\vec{x},\vec{x})d\vec{x} = \frac{\gamma_i}{8\pi\sigma^4} 12\sigma_i^4(a_i^{-4} + a_i^4 - 2)$$
$$= \frac{3\gamma_i}{2\pi}(a^{-2} - a^2)^2. \tag{3.28}$$

Since convergenc will depend upon the total residual, γ_i plays no role in the rate of convergence. Since $|a^{-1} - a| \leq 2R_{\mathcal{E}} = O(l^2)$ and if we make the reasonable restriction that $\gamma_i = O(l^2)$, we can see that the contribution from the third differential tensor is $O(l^4)$, proving the following theorem.

THEOREM 1. *If $\vec{u}$ and its derivatives are specified and bounded, the residual from the vorticity dynamics operator, (3.2), induced by an elliptical Gaussian basis function acting under system (3.12) is $O(l^4)$.*

We have established in this section that deforming basis functions induce fourth order spatial errors over the entire domain. This error estimate using residuals is not restricted to a particular norm, so that even uniform error estimates are possible as they were for CCSVM. We have also eluded to the necessity of a refinement procedure that will be discussed in §4 so that basis functions will not spread beyond a specified width l.

3.4. Integral Invariants. The vorticity dynamics equations (3.2,3.3) have many known integral invariants, so one hopes that the computational method also preserves these properties. Of course, a convergent method will satisfy these properties approximately, but some methods have the advantage of mimicking the physical structure of solutions to Navier-Stokes (3.2,3.3) by capturing certain integral invariants exactly. Unfortunately, ECCSVM does not have the property of exactly preserving first or second integral moments. Since merging and refinement (see §4) preserve all first and second moments, only the dynamics of ECCSVM will affect the integral invariants of the approximation.

Satisfying the first moment of vorticity is a problem for vortex methods using blobs of dissimilar width [12] though for corrected core spreading methods these discrepancies are controlled automatically by the parameter α because it both restricts the variation in σ_i and bounds a_i close to unity. Solutions of Navier-Stokes also satisfy satisfies the relation

$$\frac{d}{dt}\left(\iint |\vec{x}|^2\omega d\vec{x}\right) = 4\nu \iint \omega d\vec{x} \tag{3.29}$$

for the second moment of vorticity. For an elliptical Gaussian blob of unit circulation, the second moment is given by $2\sigma_i^2(a_i^2 + a_i^{-2})$. Here, indices are not necessary because this only concerns one blob. The evolution of the second moment is derived from the system (3.12).

$$\frac{d}{dt}[2\sigma^2(a^2 + a^{-2})] = 4\nu - \sigma^2(a^{-2} - a^2)\left[d_{11}(c^2 - s^2) + cs(d_{12} + d_{21})\right] \tag{3.30}$$

Of course, the first term is the exact physical integral invariant. The second term is $O(l^4)$ because $\sigma^2 \in [\alpha^2\sigma^2, \sigma^2]$ and

$$|a^{-2} - a^2| \leq 2R_{\mathcal{E}} = O(l^2). \tag{3.31}$$

The remaining term involves the flow derivatives which are assumed to be bounded.

3.5. Asymptotic Calculation of Velocity and Velocity Derivatives.. The evolution equations for the elliptical Gaussian basis functions depend upon $\vec{u}$ and $D\vec{u}$, but unlike the most other basis functions used for vortex methods, there is no known expression for these induced fields in terms of simple functions. However, the induced velocity and velocity deviations can be written as a regular perturbation in the small parameter

$$\epsilon = \frac{a-1}{a+1}, \tag{3.32}$$

expanding these field as a deviates from unity. A fourth order approximation produces excellent agreement with the exact solution in the near field, and it automatically converges in the far field where the aspect ratio is not important. Since the dynamics of ECCSVM in the appropriate convergent limit also forces $a_i^2 - 1 = (a_i - 1)(a_i + 1)$ to converge to zero, the asymptotic approximation is guaranteed to converge to the exact velocity field in the same limit.

Without loss of generality, the elliptical Gaussian element basis function is centered at the origin, and its major and minor axes are aligned with the coordinate axes ($\theta = 0$):

$$f(R;\gamma,\sigma,a) = \frac{\gamma}{4\pi\sigma^2} e^{-R^2/4\sigma^2}, \tag{3.33}$$

$$R^2 = x^2/a^2 + y^2 a^2. \tag{3.34}$$

The variable R is a continuous index of level sets of the elliptical Gaussian element. Though one could express the velocity fields as a full two dimensional Biot-Savart integral, one can also use the fact that the streamfunction (and therefore all derivatives) of an elliptical patch of vorticity with unit density and semimajor and minor axes of l_1 and l_2 can be determined using elliptical coordinates [11]:

$$\psi = \begin{cases} \frac{1}{2\pi(l_1+l_2)} \left(\frac{x^2}{l_1} + \frac{y^2}{l_2} \right), & (x,y) \in \mathcal{E}(l_1,l_2) \\ \frac{1}{2\pi} \left[\ln\left(\frac{\alpha+\beta}{l_1+l_2} \right) + \frac{\frac{x^2}{\alpha}+\frac{y^2}{\beta}}{\alpha+\beta} \right], & (x,y) \notin \mathcal{E}(l_1,l_2) \end{cases} \tag{3.35}$$

$$\alpha = \sqrt{l_1^2 + \xi} \tag{3.36}$$

$$\beta = \sqrt{l_2^2 + \xi} \tag{3.37}$$

$$1 = \left(\frac{x^2}{l_1^2 + \xi} \right) + \left(\frac{y^2}{l_2^2 + \xi} \right) \tag{3.38}$$

where $\mathcal{E}(l_1, l_2)$ is the support of the ellipse and

$$\vec{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\frac{\partial \psi}{\partial y} \\ \frac{\partial \psi}{\partial x} \end{bmatrix}$$

.

The streamfunction induced by an elliptical blob can be expressed as an infinite sum of uniform elliptical patches. For each value of R in (3.33), the constituent patch would have density $\pi R^2(-\partial_R f)dR$ and axes $l_1 = Ra$ and $l_2 = R/a$. Adding all these patches together to form an elliptical Gaussian blob, one can calculate the value of

the streamfunction at a point $(\bar{x}, \bar{y})$

$$\psi(\bar{x},\bar{y}) = -\int_0^{\bar{R}} \psi_1 \partial_R f R^2 dR - \int_{\bar{R}}^{\infty} \psi_2 \partial_R f dR \tag{3.39}$$

$$\psi_1 = \frac{1}{2}\left[\frac{\left(\frac{\bar{x}^2}{\alpha}+\frac{\bar{y}^2}{\beta}\right)}{\alpha+\beta} + \ln\left(\frac{\alpha+\beta}{Ra+R/a}\right)\right]$$

$$\psi_2 = \frac{1}{2}\frac{\frac{\bar{x}^2}{a^2}+\bar{y}^2a^2}{a+1/a}$$

$$\alpha = \sqrt{R^2a^2+\xi}$$

$$\beta = \sqrt{\frac{R^2}{a^2}+\xi} \tag{3.40}$$

Rather than using the implicit relation in equation (3.38), we can use the quadratic solution

$$\xi = \frac{1}{2}\left\{\bar{\rho}^2 - R^2\left(a^2+\frac{1}{a^2}\right) + \sqrt{\left[R^2\left(a^2+\frac{1}{a^2}\right)-\bar{\rho}^2\right]^2 + 4R^2(\bar{R}^2-R^2)}\right\} \tag{3.41}$$

where $\bar{\rho}^2 = \bar{x}^2+\bar{y}^2$ and $\bar{R}^2 = \frac{\bar{x}^2}{a^2}+\bar{y}^2a^2$. Important limits to be used later are $\xi \to \bar{\rho}^2$ as $R \to 0$ and $\xi \to 0$ as $R \to \bar{R}$.

While ψ_2 is constant with respect to R making the second term on the right side of (3.39) elementary to integrate, ψ_1 is not readily expressed in terms of the variable R making the first term problematic. However, one can approximate ψ_1 in powers of the small parameter ϵ. Expanding in ϵ rather than R has the advantage that ϵ can be made to be a truly small parameter. The parameter R is not guaranteed to be small relative to σ since one may wish to use the streamfunction virtually anywhere in the domain, so the series may converge slowly. Also, expanding in ϵ will yield a uniformly valid approximation of ψ_1 over the entire domain. The elliptical radius, R, is the natural variable of integration. At the same time, $\bar{R}$ is a parameter representing one possible combination of $\bar{x}$ and $\bar{y}$. One of many possible complements is $\bar{T}$ where $\bar{T}^2 = \frac{\bar{x}^2}{a^2} - \bar{y}^2a^2$. Combined with the fact that the streamfunction must retain a twofold symmetry about the major and minor axes, the streamfunction can be expressed as a function of $\bar{R}$ and $\bar{T}$. One could also use coordinates radially symmetric coordinates such as $\bar{x}^2+\bar{y}^2$ and $\bar{x}^2-\bar{y}^2$, but it was found that these parameters yields less accurate streamfunction approximations because they do not explicitly contain information about the elliptical geometry. For instance, a term involving $\bar{x}^2+\bar{y}^2$ must be corrected at the next order of ϵ to express information about the elliptical radius.

From (3.32), we can use

$$a = \frac{1+\epsilon}{1-\epsilon}, \tag{3.42}$$

to find that

$$\bar{\rho}^2 = \bar{R}^2 + 4\epsilon\bar{S}^2 - 8\epsilon^2\bar{R}^2 + -20\epsilon^3\bar{S}^2 + 48\epsilon^4\bar{R}^2 + O(\epsilon^5), \tag{3.43}$$

where $\bar{S}^2 = \bar{x}^2 - \bar{y}^2$, so that

$$\bar{S}^2 = \bar{T}^2 + 4\epsilon\bar{R}^2 + 8\epsilon^2\bar{T}^2 + 12\epsilon^3\bar{R}^2 + 16\epsilon^4\bar{T}^2 + O(\epsilon^5), \tag{3.44}$$

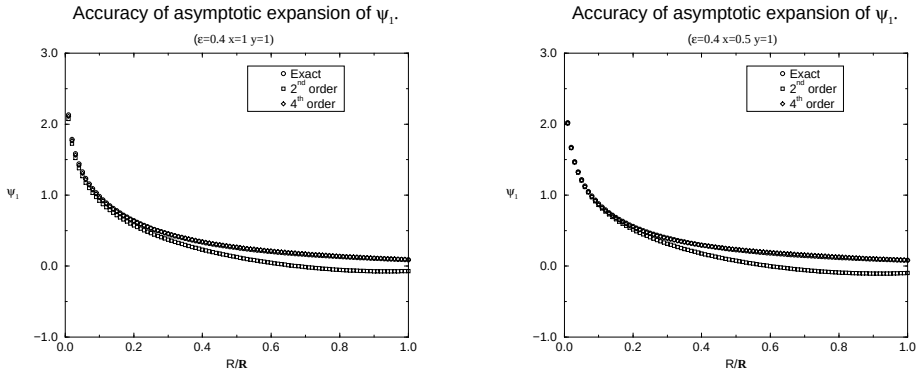

FIG. 3.1. *A example of the accuracy of the fourth order asymptotic expansion of* Ψ_1*. A second-order (squares) and fourth-order (diamonds) expansion are shown together with the exact function (circles). The aspect ratio of 5.44... (*$\epsilon = 0.4$*) is chosen intentionally to be large to demonstrate that the fourth order approximation is quite robust. The streamfunction is calculated with* $\bar{x} = 1$ *and* $\bar{y} = 1$ *on the left and* $\bar{x} = \frac{1}{2}$ *and* $\bar{y} = 1$ *on the right.*

and therefore

$$\bar{\rho}^2 = \bar{R}^2 + 4\epsilon\bar{T}^2 + 8\epsilon^2\bar{R}^2 + 12\epsilon^3\bar{T}^2 + 16\epsilon^4\bar{R}^4 + O(\epsilon^5). \tag{3.45}$$

Substituting the expression above into the streamfunctions (3.40) and (3.41) and expanding in powers of ϵ, one obtains

$$\begin{aligned}
\psi_1 = &\frac{1}{4}\left[1 + \ln\left(\frac{\bar{R}^2}{R^2}\right)\right] + \epsilon\left(\frac{\bar{T}^2}{\bar{R}^2} - \frac{1}{2}\frac{\bar{T}^2R^2}{\bar{R}^4}\right) + \\
&\epsilon^2\left[1 - 2\frac{\bar{T}^4}{\bar{R}^4} + \left(-2 + 4\frac{\bar{T}^4}{\bar{R}^4}\right)\frac{R^2}{\bar{R}^2} + \left(1 - 2\frac{\bar{T}^4}{\bar{R}^4}\right)\frac{R^4}{\bar{R}^4}\right] + \\
&\epsilon^3\left[\frac{16}{3}\frac{\bar{T}^6}{\bar{R}^6} - 5\frac{\bar{T}^2}{\bar{R}^2} + \left(\frac{37}{2}\frac{\bar{T}^2}{\bar{R}^2} - 24\frac{\bar{T}^6}{\bar{R}^6}\right)\frac{R^2}{\bar{R}^2} + \left(-24\frac{\bar{T}^2}{\bar{R}^2} + 32\frac{\bar{T}^6}{\bar{R}^6}\right)\frac{R^4}{\bar{R}^4} + \right. \\
&\left.\left(10\frac{\bar{T}^2}{\bar{R}^2} - \frac{40}{3}\frac{\bar{T}^6}{\bar{R}^6}\right)\frac{R^6}{\bar{R}^6}\right] + \\
&\epsilon^4\left[-4 + 20\frac{\bar{T}^4}{\bar{R}^4} - 16\frac{\bar{T}^8}{\bar{R}^8} + \left(20 - 136\frac{\bar{T}^4}{\bar{R}^4} + 128\frac{\bar{T}^8}{\bar{R}^8}\right)\frac{R^2}{\bar{R}^2} + \right. \\
&\left(-42 + 324\frac{\bar{T}^4}{\bar{R}^4} - 320\frac{\bar{T}^8}{\bar{R}^8}\right)\frac{R^4}{\bar{R}^4} + \left(40 - 320\frac{\bar{T}^4}{\bar{R}^4} + 320\frac{\bar{T}^8}{\bar{R}^8}\right)\frac{R^6}{\bar{R}^6} + \\
&\left.\left(-14 + 112\frac{\bar{T}^4}{\bar{R}^4} - 112\frac{\bar{T}^8}{\bar{R}^8}\right)\frac{R^8}{\bar{R}^8}\right] + O(\epsilon^5) \\
\psi_2 = &\frac{1}{2}\left[\frac{1}{2}\bar{R}^2 + \epsilon\bar{T}^2 - \epsilon^3\bar{T}^2\right] + O(\epsilon^5).
\end{aligned} \tag{3.46}$$

Since ψ_2 does not vary with R, the main concern is that ψ_1 remain accurate over $R \in [0, \bar{R}]$.

In addition to being an excellent approximation to the exact streamfunction (see Fig. (3.1)), the asymptotic approximation (3.46) has two important advantages. First, the fourth order approximation in ϵ (3.46) *can* be integrated exactly in equation (3.39) because derivatives of (3.46) involve only even powers of R. Therefore, calculating

velocities and velocity deviations is simply a matter of differentiating the exact expression with respect to $\bar{x}$ and $\bar{y}$. No term from the "Leibnitz rule" is retained when differentiating because ψ is continuous and continuously differentiable across the boundary at $R = \bar{R}$. Second, incompressibility is exactly preserved both locally and globally using the approximate streamfunction as would not be case if one were to numerically approximate the Biot-Savart integrals.

Observing that the fourth order approximation for the streamfunction involves only zeroth, second, fourth and sixth moments of $\partial_R f$, we anticipate that all integrals approximating velocities, $u = -\partial_y \psi$, $v = \partial_x \psi$, as well as velocity deviations $\partial_x u = -\partial_{xy}\psi$, etc. will involve integrals of the form,

$$\begin{aligned} I &= -\int_0^{\bar{R}} R^2(C_0 + C_1 R^2 + C_2 R^4 + C_3 R^6 + C_4 R^8)\partial_R f dR - C_5 f(\bar{R}) \\ &= \frac{\gamma}{2\pi}\left\{2C_0 + 16C_1\sigma^2 + 192C_2\sigma^4 + 3072C_3\sigma^6 + 61440C_4\sigma^8\right. \\ &\quad - \left[C_0\left(\frac{\bar{R}^2}{2\sigma^2} + 2\right) + C_1\left(\frac{\bar{R}^4}{2\sigma^2} + 4\bar{R}^2 + 16\sigma^2\right)\right. \\ &\quad + C_2\left(\frac{\bar{R}^6}{2\sigma^2} + 6\bar{R}^4 + 48\bar{R}^2\sigma^2 + 192\sigma^4\right) \\ &\quad + C_3\left(\frac{\bar{R}^8}{2\sigma^2} + 8\bar{R}^6 + 96\sigma^2\bar{R}^4 + 768\sigma^4\bar{R}^2 + 3072\sigma^6\right) \\ &\quad + C_4\left(\frac{\bar{R}^{10}}{2\sigma^2} + 10\bar{R}^8 + 160\sigma^2\bar{R}^6 + 1920\sigma^4\bar{R}^4 + 15360\sigma^6\bar{R}^2 + 61440\sigma^8\right) \\ &\quad \left.\left. - \frac{C_5}{2\sigma^2}\right]\exp\left(-\frac{\bar{R}^2}{4\sigma^2}\right)\right\}. \end{aligned} \tag{3.47}$$

The coefficients, C_i, $(i = 0, 1, 2, 3, 4)$, have been determined after many manipulations and reductions. One can exploit symmetries in ψ and reduce the amount of work by a factor of two by storing intermediate variables c_i^{Even} and c_i^{Odd}, listed in Table (3.1). For $u = -\partial_y \psi$,

$$C_i = -\frac{\bar{y}}{2}\left(c_i^{\text{Even}} - c_i^{\text{Odd}}\right), \tag{3.48}$$

and for $v = \partial_x \psi$,

$$C_i = \frac{\bar{x}}{2}\left(c_i^{\text{Even}} + c_i^{\text{Odd}}\right). \tag{3.49}$$

Similarly, one can use the fourth order expand of ψ_1 to evaluate second spatial derivatives that are necessary to calculate flow deviations. The necessary additional intermediate coefficients are tabulated in Table (3.2). For $\partial_y u = -\partial_{yy}\psi$,

$$C_i = -\frac{1}{2}\left(c_i^{\text{Even}} - c_i^{\text{Odd}}\right) - \frac{\bar{y}^2}{\bar{R}^2}\left(d_i^{\text{Even}} - d_i^{\text{Odd}}\right), \tag{3.50}$$

and for $\partial_x v = \partial_{xx}\psi$,

$$C_i = \frac{1}{2}\left(c_i^{\text{Even}} + c_i^{\text{Odd}}\right) + \frac{\bar{x}^2}{\bar{R}^2}\left(d_i^{\text{Even}} + d_i^{\text{Odd}}\right). \tag{3.51}$$

Function	
c_0^{Even}	$\frac{1}{\bar{R}^2}\left[1-4\epsilon\frac{\bar{T}^2}{\bar{R}^2}+\epsilon^2\left(-8+16\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^3\left(52\frac{\bar{T}^2}{\bar{R}^2}-64\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^4\left(48-288\frac{\bar{T}^4}{\bar{R}^4}+256\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
c_0^{Odd}	0
c_1^{Even}	$\frac{\epsilon}{\bar{R}^4}\left[4\frac{\bar{T}^2}{\bar{R}^2}+\epsilon\left(16-48\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^2\left(-244\frac{\bar{T}^2}{\bar{R}^2}+384\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^3\left(-288+2400\frac{\bar{T}^4}{\bar{R}^4}-2560\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
c_1^{Odd}	$\frac{\epsilon}{\bar{R}^4}\left[-2+\epsilon 16\frac{\bar{T}^2}{\bar{R}^2}+\epsilon^2\left(26-96\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^3\left(-288\frac{\bar{T}^2}{\bar{R}^2}+512\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
c_2^{Even}	$\frac{\epsilon^2}{\bar{R}^6}\left[-8+32\frac{\bar{T}^4}{\bar{R}^4}+\epsilon\left(352\frac{\bar{T}^2}{\bar{R}^2}-640\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^2\left(656-6464\frac{\bar{T}^4}{\bar{R}^4}+7680\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
c_2^{Odd}	$\frac{\epsilon^2}{\bar{R}^6}\left[-16\frac{\bar{T}^2}{\bar{R}^2}+\epsilon\left(-64+256\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^2\left(1312\frac{\bar{T}^2}{\bar{R}^2}-2560\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
c_3^{Even}	$\frac{\epsilon^3}{\bar{R}^8}\left[\left(-160\frac{\bar{T}^2}{\bar{R}^2}+320\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon\left(-640+7040\frac{\bar{T}^4}{\bar{R}^4}-8960\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
c_3^{Odd}	$\frac{\epsilon^3}{\bar{R}^8}\left[\left(40-160\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon\left(-1920\frac{\bar{T}^2}{\bar{R}^2}+3840\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
c_4^{Even}	$\frac{\epsilon^4}{\bar{R}^{10}}\left(224-2688\frac{\bar{T}^4}{\bar{R}^4}+3584\frac{\bar{T}^8}{\bar{R}^8}\right)$
c_4^{Odd}	$\frac{\epsilon^4}{\bar{R}^{10}}\left(896\frac{\bar{T}^2}{\bar{R}^2}-1792\frac{\bar{T}^6}{\bar{R}^6}\right)$
c_5^{Even}	1
c_5^{Odd}	$-2\epsilon+2\epsilon^3$

TABLE 3.1
Table of coefficients for velocity calculations

For $\partial_y v = -\partial_x u = \partial_{xy}\psi$, some intermediate variables can be recycled once again.

$$\begin{aligned} C_0 &= \frac{\bar{x}\bar{y}}{\bar{R}^2} d_0^{\text{Even}} \\ C_1 &= \frac{\bar{x}\bar{y}}{\bar{R}^2} d_1^{\text{Even}} \\ C_2 &= \frac{\bar{x}\bar{y}}{\bar{R}^2} e_2 \\ C_3 &= \frac{\bar{x}\bar{y}}{\bar{R}^2} e_3 \\ C_4 &= \frac{\bar{x}\bar{y}}{\bar{R}^2} e_4 \\ C_5 &= 0 \end{aligned} \tag{3.52}$$

While a second order expansion in ϵ was not sufficiently close to the streamfunction (see Fig. (3.1)) for the calculation of either first or second derivatives, Fig. (3.2) demonstrates that the fourth order approximation yields very accurate results.

4. Adaptive Refinement (Splitting Elements). Spatial refinement plays the same role for ECCSVM as it does for CCSVM. In CCSVM, a Gaussian blob which has spread beyond a prescribed spatial resolution l would be replaced an overlapping configuration of "thinner" blobs. The new configuration would approximate the original element in same sense whether it is minimizing errors or conserving moments. One should observe that the original refinement procedure for CCSVM only conserved a

Function	
d_0^{Even}	$\frac{1}{\bar{R}^2}\left[-1+8\epsilon\frac{\bar{T}^2}{\bar{R}^2}+\epsilon^2\left(16-48\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^3\left(-168\frac{\bar{T}^2}{\bar{R}^2}+256\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^4\left(-160+1248\frac{\bar{T}^4}{\bar{R}^4}-1280\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
d_0^{Odd}	0
d_1^{Even}	$\frac{\epsilon}{\bar{R}^4}\left[-12\frac{\bar{T}^2}{\bar{R}^2}+\epsilon\left(-48+192\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^2\left(1020\frac{\bar{T}^2}{\bar{R}^2}-1920\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^3\left(1248-12672\frac{\bar{T}^4}{\bar{R}^4}+15360\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
d_1^{Odd}	$\frac{\epsilon}{\bar{R}^4}\left[8-\epsilon 96\frac{\bar{T}^2}{\bar{R}^2}+\epsilon^2\left(-168+768\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^3\left(2496\frac{\bar{T}^2}{\bar{R}^2}-5120\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
d_2^{Even}	$\frac{\epsilon^2}{\bar{R}^6}\left[8-160\frac{\bar{T}^4}{\bar{R}^4}+\epsilon\left(-1408\frac{\bar{T}^2}{\bar{R}^2}+3840\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^2\left(-2768+36160\frac{\bar{T}^4}{\bar{R}^4}-53760\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
d_2^{Odd}	$\frac{\epsilon^2}{\bar{R}^6}\left[128\frac{\bar{T}^2}{\bar{R}^2}+\epsilon\left(512-2560\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon^2\left(-13568\frac{\bar{T}^2}{\bar{R}^2}+30720\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
d_3^{Even}	$\frac{\epsilon^3}{\bar{R}^8}\left[\left(480\frac{\bar{T}^2}{\bar{R}^2}+-2240\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon\left(1920-38400\frac{\bar{T}^4}{\bar{R}^4}+71680\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
d_3^{Odd}	$\frac{\epsilon^3}{\bar{R}^8}\left[\left(-320+1920\frac{\bar{T}^4}{\bar{R}^4}\right)+\epsilon\left(21760\frac{\bar{T}^2}{\bar{R}^2}-53760\frac{\bar{T}^6}{\bar{R}^6}\right)\right]$
d_4^{Even}	$\frac{\epsilon^4}{\bar{R}^{10}}\left(-224+13440\frac{\bar{T}^4}{\bar{R}^4}-32256\frac{\bar{T}^8}{\bar{R}^8}\right)$
d_4^{Odd}	$\frac{\epsilon^4}{\bar{R}^{10}}\left(-10752\frac{\bar{T}^2}{\bar{R}^2}+28672\frac{\bar{T}^6}{\bar{R}^6}\right)$
d_5^{Even}	0
d_5^{Odd}	0
e_2	$\frac{\epsilon^2}{\bar{R}^6}\left[40-160\frac{\bar{T}^4}{\bar{R}^4}+\epsilon\left(-1920\frac{\bar{T}^2}{\bar{R}^2}+3840\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon^2\left(-3600+41280\frac{\bar{T}^4}{\bar{R}^4}-53760\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
e_3	$\frac{\epsilon^3}{\bar{R}^8}\left[\left(1120\frac{\bar{T}^2}{\bar{R}^2}-2240\frac{\bar{T}^6}{\bar{R}^6}\right)+\epsilon\left(4480-53760\frac{\bar{T}^4}{\bar{R}^4}+71680\frac{\bar{T}^8}{\bar{R}^8}\right)\right]$
e_4	$\frac{\epsilon^4}{\bar{R}^{10}}\left(-2016+24192\frac{\bar{T}^4}{\bar{R}^4}-32256\frac{\bar{T}^8}{\bar{R}^8}\right)$

TABLE 3.2
Table of coefficients for velocity deviation calculations

composite second moment ($\int\int (x^2+y^2)f(x,y)dxdy$). In the case of elliptical Gaussian basis functions, it is possible to conserve *all* second moments. Necessarily, any refinement process must reduce the core width to maintain consistency.

Adaptive refinement for ECCSVM with four elements is a direct adaptation of adaptive refinement for CCSVM [22]. Rather than arranging the configuration uniformly around the element centroid, one arranges the elements on an ellipse around the centroid. Without loss of generality, the arrangement can be considered to have its centroid at the origin and its principal axes aligned with the cardinal axes. Using (3.9) with $\theta = 0$ for can examine the difference between an element and four thinner replacements:

$$\begin{aligned}\delta\widehat{\omega} = g(\vec{x};\gamma,(0,0),\sigma,a,0)-\\ [g(\vec{x};\gamma/4,(r\sigma/a,0),\alpha\sigma,a,0)+g(\vec{x};\gamma/4,(-r\sigma/a,),\alpha\sigma,a,0)+\\ g(\vec{x};\gamma/4,(0,ra\sigma),\alpha\sigma,a,0)+g(\vec{x};\gamma/4,(0,-ra\sigma),\alpha\sigma,a,0)]\end{aligned} \quad (4.1)$$

where α is a controllable parameter governing the accuracy of the refinement process and r must be determined. If one is trying to conserve moments, one of the advantages

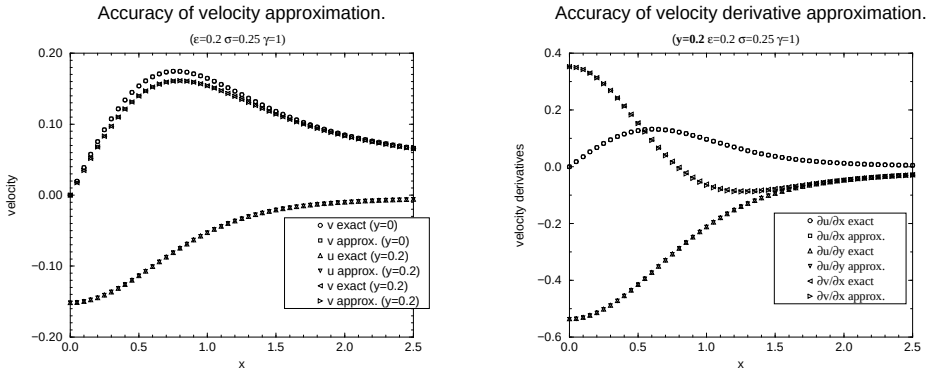

FIG. 3.2. *Sample velocity (left) and velocity derivative (right) calculations for an elliptical Gaussian element. In this case,* $\epsilon = 0.2$ *corresponding to an aspect ratio of 2.25. One can see that corresponding symbols cover one another indicating a very accurate match between the asymptotic approximation and the "exact" quantity that is calculated numerically to machine precision.*

of this procedure using four elements is that it preserves the aspect ratio of the original element. This system automatically conserves zeroth and first moments by construction. The symmetries built into this form also allow one to conserve all second moments with an appropriate choice of $r = 2\sigma\sqrt{(1-\alpha^2)}$.

While conserving moments was the approach originally proposed by Rossi in [22], Shiels studied refinement to a greater extent in [26]. Shiels tried a variety of different configurations other than one to four, but found that none substantially affected the induced refinement error. Shiels observed that combinations and cancellations during an extended simulation would affect the cumulative affect of refinement. Shiels performed an exhaustive search for an optimal function, $r = f(\alpha)\sigma$, that would minimize the cumulative refinement error through a series of refinement events. He computed $f(\alpha)$ by collecting optimal values of f in a number of Lamb monopole simulations, and from his data for the interval $0.6 \leq \alpha \leq 0.8$, he proposed a linear model:

$$f(\alpha) = 3.9 - 3.6\alpha.$$

Unfortunately, accumulations and cancellations are strongly dependent on the flow structure, so that what might be optimal for one flow, may not be for another. Also, this particular linear model is inconsistent as $\alpha \to 1$.

At least at this point, local analysis of a single refinement event is the best way to proceed. Conserving second moments,

$$f(\alpha) = 2\sqrt{1-\alpha^2},$$

may not result is the least induced refinement error. This is true both instantaneously (see Fig. (4.1)) and as a cumulative effect for the Lamb-Oseen monopole [26]. However, it is a consistent means of determining $f(\alpha)$. Fig. (4.1) compares a numerically-computed optimal curve to both the linear model and the model which conserves the second moment. To find a consistent fit to the optimal computation, we seek a function the form

$$f(\alpha) = \sqrt{(1-\alpha)(C_1\alpha + C_2)}.$$

The best (least squares) fit to this curve using the data in Fig. (4.1) is

$$f(\alpha) = \sqrt{(1-\alpha)(9.7052\alpha - 1.7235)} \tag{4.2}$$

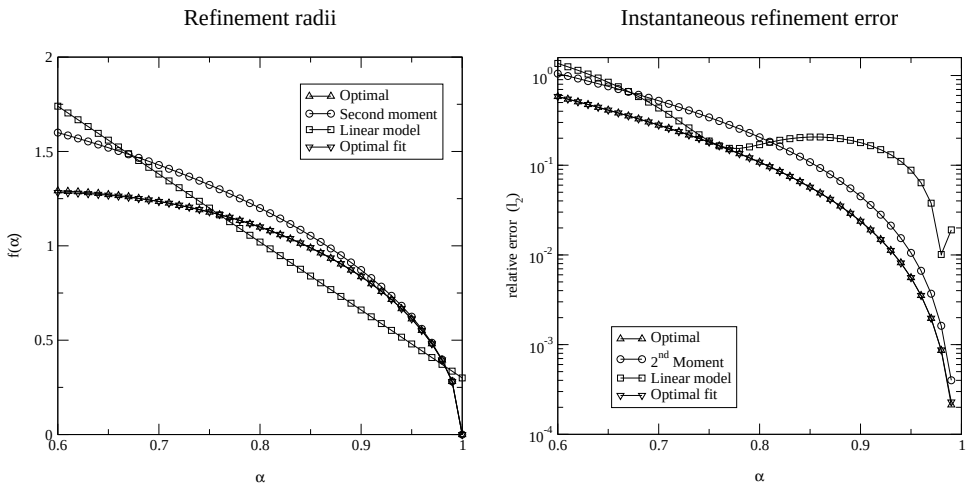

FIG. 4.1. *Refinement radius (left) and instantaneous induced errors (right) for 1 → 4 refinement. The "optimal" curve is calculated numerically to minimize the induced l_2 error for a single splitting event. The second moment curve converges to the optimal curve as $\alpha \to 1$. The linear model is inconsistent. However, it should also be noted that Sheils only performed his computations over a limited range of α from 0.6 to 0.8, and in this subrange, the linear model outperforms the conservative second moment radii.*

provides an excellent fit to the numerically computed curve as seen in Fig. (4.1).

With the exception of the linear model, the error induced by a single refinement event will go to zero as $\alpha \to 1$. However, the number of refinements will grow as $\alpha \to 1$ because elements are not being refined as much. Knowing that $\frac{1}{2}(a^2 + a^{-2}) \geq 1$, one can see from (3.12) that if Δt is the interval between refinement events,

$$\Delta t \leq \frac{l^2(1-\alpha^2)}{\nu}. \tag{4.3}$$

Rossi analyzed the second moment method as applied to CCSVM and determined its convergence rate over a fixed duration for any α to be $O(l^2) + O[l(1-\alpha^2)]$. While the same method of analysis is not applicable to ECCSVM, the procedure is similar enough that one would expect similar convergence behavior although the question remains open.

5. Sample Calculations. The spatial convergence rate of ECCSVM is demonstrated by using it to compute the flow induced by an Lamb-Oseen monopole which has known exact solutions. We choose one corresponding to a circulation of π and an initial core size of $1/4$:

$$\omega(\vec{x}, t) = \frac{1}{4(\frac{1}{16} + \nu t)} \exp\left[-\frac{|\vec{x}|^2}{4(\frac{1}{16} + \nu t)}\right] \tag{5.1}$$

Here, we choose the viscosity to be $\nu = \frac{1}{100}$. While ECCSVM maintains its accuracy using spatial refinement, we shall intentionally integrate the flow over a short enough time to avoid refinement and therefore avoid refinement-induced errors. By removing this effect, we eliminate all sources of error except spatial error and temporal integration error. To gauge nonlinear effects, we perform each computation both in a linear regime where the exact velocity is specified, and a nonlinear regime where the flow is computed via the Biot-Savart integration.

Each flow is integrated out to $T = 0.1$ which corresponds to about $1/30^{\text{th}}$ of a turnover with a timestep of 2.5×10^{-3} for a total of 40 steps. The system (3.12) is integrated using a split step scheme. The particle trajectories are integrating using a third order Adams-Bashforth scheme. The width, aspect ratio and orientation advance using a fourth order Runge-Kutta scheme that is dependent upon a second order midpoint velocity field computation. As usual for vortex methods, it is the velocity field which dominates the computational effort.

Each initial condition was generated by seeding the flow with basis functions of initial core size σ_0^2 and aspect ratio $a_0^2 \neq 1$. The latter is necessary because we know that the total spatial error is $O(l^4) + O[(a_M^2 - a_M^{-2})^2]$. While it is true that $a_M^2 - a_M^{-2}$ scales with l^2 over time, if one used $a_0^2 = 1$ as an initial condition, one would have to integrate the flow for long periods of time to wait for the aspect ratios to reach their equilibrium values. To initiate the simulation with exact initial data and specified initial core areas and aspect ratios, we use the following convolution:

$$\frac{1}{4\pi\sigma^2} \exp\left[-\frac{|\vec{x}|^2}{4\sigma^2}\right] = h(a_1, \sigma_1) * h\left(\left[\frac{\sigma^2 - \sigma_1^2 a_1^2}{\sigma^2 - \sigma_1^2/a_1^2}\right]^{\frac{1}{4}}, \left[\sigma^4 + \sigma_1^4 - \sigma^2\sigma_1^2(a_1^2 + a_1^{-2})\right]^{\frac{1}{4}}\right) \quad (5.2)$$

where

$$h(a, \sigma) = \frac{1}{4\pi\sigma^2} \exp\left[-\frac{x^2/a^2 + y^2 a^2}{4\sigma^2}\right]. \quad (5.3)$$

This convolution is related to the exact deregularization discussed in [24] for perturbed Gaussian monopoles. Using this deregularization, one can choose the initial width and aspect ratio of the basis functions, and use them to express the desired initial condition ((5.1) at $t = 0$), to a precision which is substantially greater than the spatial and temporal errors in the flow computation. For each simulation, the initial aspect ratio is chosen to be a value close to the theoretical a_M^2. Even though a_M^2 varies over the spatial domain, only one value may be chosen to use (5.3), so a typical value is selected. The overlap between elements is fixed to maintain a constant discretization error for all simulations.

The Lamb-Oseen vortex was simulated over a range of core areas varying over a decade to validate the fourth order accuracy of ECCSVM. The problem size increases roughly 32-fold as the spatial errors are reduced by a factor of approximately 200. For the nonlinear computations, I implemented a fast multipole solver based on Greengard and Rokhlin's algorithm, but modified substantially for ECCSVM [4]. For the nonlinear studies, computation times varied from several minutes at low resolution to over an hour for high resolution runs on an SGI MIPS R12000 using one processor.

The results of the simulations, as displayed in Fig. (5.1), show that ECCSVM is a fourth order method in full agreement with the linear analysis for both the linear and nonlinear simulations. Since the core size does not remain constant during the simulation, we use the approximate average core size

$$\sigma_0^2 + \frac{1}{2}\nu T$$

to represent the cumulative effect of the spatial errors.

σ_0^2	a_0^2	N
1.28×10^{-2}	3.0	771
6.4×10^{-3}	2.0	1,723
3.2×10^{-3}	1.5	3,431
1.6×10^{-3}	1.25	6,939
8.0×10^{-4}	1.125	13,893
4.0×10^{-4}	1.0625	26,317

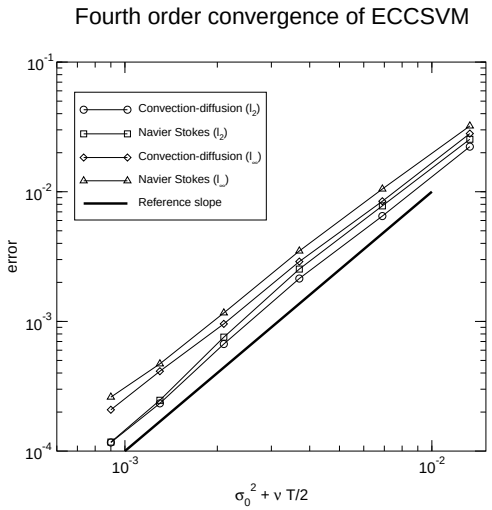

FIG. 5.1. *Uniform and l_2 convergence of ECCSVM for Lamb-Oseen dipole. The initial core areas, aspect ratios and problem sizes (left) are computed in tandem based on equilibrium values for the Lamb-Oseen monopole. The convergence rate of ECCSVM (left) is demonstrated over a decade of core areas. Both l_2 and uniform (l_∞) errors are displayed near the reference line which has a slope of $m = 2$.*

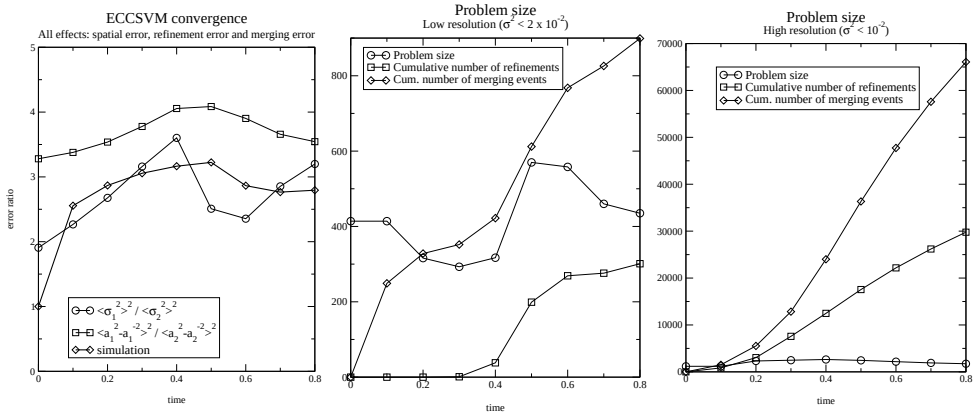

FIG. 5.2. *Convergence of ECCSVM over a modest period of time with all effects included. On the left, weighted means such as $< \sigma^2 >= \sum \gamma_i \sigma_i^2$ can be used to estimate the two dominant terms in the spatial error. The subscripts 1 and 2 refer to low ($l = 2 \times 10^{-2}, \alpha = 0.8$) and high resolution ($l = 10^{-2}, \alpha = 0.95394$) simulations, respectively. The refinement parameter, α is chosen to force the refinement errors to match a fourth order spatial error reduction. At the center and right, one sees that merging can stabilize the growth in problem size. In these measurements, no information is presented about the nature of the merging event, and one should know that a single merging event can involve many elements being collapsed into a single one. Elements are merged at regular intervals, and the merger-induced errors are not large relative to spatial errors and refinement errors for the time series shown.*

In practice, a vortex computation will include spatial refinement and some form of problem size control whether it is merging or some other technique. To gauge the impact of these effects, I computed the statistics for the Lamb-Oseen computation over a modest duration during which time computational elements would undergo refinement and merging. Aside from errors from the temporal integration, there are

spatial errors, refinement-induced errors and merging errors. The first two have been discussed in some detail in this paper. In Fig. (5.2), one can see that high spatial accuracy can be maintained in the presense of refinement and merging if the parameters are chosen appropriately. (For instance, if one does not reduce the refinement parameter, α, refinement errors will dominate the computation and there is no point in trying to improve the spatial resolution.) In stabilizing the problem size, one would expect that merging error would eventually swamp the computation after a long time, but up to that point, spatial and refinement errors would determine the accuracy of the vortex method. In the figure, one can see that true fourth order behavior is never realized. The reason is not that merging or refinement has generated great-than-expected errors in the flow. Rather, the *low* resolution computation is performing *better* than one would expect because $(\alpha l)^2$ for the low resolution experiment is not precisely twice the size of $(\alpha l)^2$ for the high resolution experiment. This is shown clearly in Fig. (5.2) sometime between $t = 0.4$ and $t = 0.5$. Initially, the performance of ECCSVM climbs as the low resolution core sizes grow, but the high resolution cores remain constrained in a narrow range through steady refinement and merging. Sometime between $t = 0.4$ and $t = 0.5$, most of the elements in the low resolution experience undergo refinement, effectively dropping their widths back down to αl and similarly dropping the ratio of core sizes. After this point, we see the performance ratio drop, as we would expect, not from any source of error in the high resolution experiment but rather because elements in the low resolution experiment are not wide all the time.

6. Open Source Distribution. This algorithm has been implemented in the programming language C, and scaled for parallel operations using the Message Passing Interface (MPI). This implementation along with a variety of related tools is available at no cost at the following World Wide Web sites:

`http://sherlock.math.uml.edu/BlobFlow`

`http://www.math.udel.edu/~rossi/BlobFlow`

The simulations in this paper were performed with BlobFlow version 2.0. The linearized calculations were generated with a very slight modification.

The open distribution of source code for complex algorithms such as ECCSVM is in the public interest for several reasons.

1. Open source distribution of complex algorithms like these saves investigators considerable time and effort coding from manuscripts into a programming language. For instance, coding and debugging the asymptotic approximation to the Biot-Savart integral for elliptical Gaussian basis functions requires a substantial investment of time that is unnecessary if an open source implementation is available.
2. The distribution of source code allows other investigators to reproduce the experiments and examples in this paper, and examine them in greater detail, allowing more intensive study of the topics covered in this manuscript and related manuscripts.
3. Open source distribution makes the algorithm easily available to the scientific community for uses other than those discussed in this paper. In fact, the widespread availability of modifiable source code permits the broader application of the general algorithm to other areas of research not specifically envisioned by the author.
4. A community of mathematicians and scientists working with an open source distribution can more quickly modify, optimize and extend the current im-

plementation than a single person.
There is substantial discussion of open source software in the information technology fields mostly centered around intellectual property, reliability and interoperability (see [5, 6, 20, 30] for example,) but relatively little in the computational mathematics community where reproducibility and scrutiny is paramount. All the parameter files used the generate the data discussed in this article are available from the author. The author hopes that making the program open to the entire scientific community will result in better mathematics, better science and better tools for investigators.

7. Concluding Remarks and Future Work. The use of deforming elliptical Gaussian basis functions for Lagrangian schemes has yielded a naturally adaptive scheme with a high spatial accuracy. More than anything else, the underlying motivation is to boost the accuracy of Lagrangian schemes without sacrificing the compactness of the basis function. ECCSVM achieves this result for fully viscous calculations, but the per element computational cost for the Biot-Savart integration is higher than for axisymmetric basis functions. In practice, ECCSVM uses a fast multipole summation routine, so the high cost of the Biot-Savart calculation only affects direct interactions, not the far-field calculations.

This introduces numerous avenues for further exploration, both by improving the numerics for better accuracy and performance, and by extending the method to incorporate more physical effects. All vortex methods would be improved with high accuracy time integration routines. With fourth order spatial accuracy, one would hope to find a fourth order method in time as well. In ECCSVM, we see that the trajectories are expensive to integrate, but the evolution of the internal parameters such as σ or a is relatively cheap to calculate. In general, this is typical for all vortex methods: The velocity field computation uses most of the resources while regridding, viscous diffusion and other corrections are cheap. Since velocity calculations are expensive, most investigators use "Adams family" multistep methods, but typically one is forced to operate at a low order because the computational elements are interrupted at regular intervals by refinement, regridding or other necessary processes. Also, a higher order refinement procedure would improve ECCSVM substantially. In §5, we see that α must be increased substantially to match the spatial order of the method, and so at high resolutions, a tug-of-war develops between refinement and merging.

Adding constraints for solid boundaries using panel methods would be a simple extension, and the elliptical basis functions will resolve boundary layers more efficiently than radially symmetric boundary layers as Marshall, Grant and Teng have found [17, 27]. More challenging still would be extending this method to three dimensions. Though increasing dimensionality adds several new challenges, particularly in evaluating the streamfunction, these problems are accessible using asymptotics similar to those employed here.

Finally, we see that this high order method is a new tool for exploring vorticity dynamics in high Reynolds number flows, and not only does the method work well, but the linear analysis of the residual accurately describes the actual convergence properties of the method. The improved spatial resolution and adaptivity make this scheme a natural choice for problems involving vorticity-dominated flows. At the same time, improving the spatial accuracy by introducing deforming computational elements in vortex methods opens the door to a variety of interesting extensions and improvements to the method that have yet to be fully explored.

8. Acknowledgments. The author thanks Chad Fox and the anonymous referee for many useful questions and comments. The author would like to thank all

those who have supported this effort through many dead-ends and frustrating moments including TAR, CTR, JFL, AJB and those for whom squash is more than a vegetable.

Appendix A. Derivation of Elliptical Gaussian Basis Function Dynamics.

We begin by choosing the origin to the center of the basis function and orienting (3.9) along the principal axes ($\theta = \theta_i$) which we shall refer to as X and Y. We force this coordinate system to satisfy the local equation (3.7). In the interest of brevity, all indices i are dropped and we focus on a single basis function. As before, we denote the X and Y components of $\vec{u}$ as $u^{(1)}$ and $u^{(2)}$ respectively. Expanding in powers of X and X, we can equate each coefficient with zero to derive the appropriate dynamical system, and set $\theta = 0$. We drop all indices except on $\vec{x}$ and $\vec{x}_i$ to distinguish between the independent variable and the position of the vortex element.

$$O(1) : -\frac{1}{\sigma^2}\frac{d}{dt}(\sigma^2) + \nu\frac{\left(a^{-2}+a^2\right)}{2\sigma^2} = 0 \tag{A.1}$$

$$O(X) : a^{-2}(\frac{d}{dt}x_i - u^{(1)}) = 0 \tag{A.2}$$

$$O(Y) : a^2(\frac{d}{dt}y_i - u^{(2)}) = 0 \tag{A.3}$$

$$O[X^2] : \frac{d}{dt}(\sigma^2)a^{-2} + \sigma^2 a^{-4}\frac{d}{dt}(a^2) - 2\sigma^2 d_{11}a^{-2} - \nu a^{-4} = 0 \tag{A.4}$$

$$O[Y^2] : -\frac{d}{dt}(\sigma^2)a^2 - \sigma^2\frac{d}{dt}(a^2) + 2\sigma^2 d_{11}a^2 - \nu a^4 = 0 \tag{A.5}$$

$$O[XY] : -\sigma^2\frac{d\theta}{dt}(a^{-2} - a^2) - \sigma^2[d_{12}a^{-2} + d_{21}a^2] = 0 \tag{A.6}$$

Clearly, the evolution of the basis function width in (3.12) arises from (A.1):

$$\frac{d\sigma^2}{dt} = \nu(a^2 + a^{-2}).$$

The obvious evolution equations for position, $\vec{x}_i$ in (3.12), come from (A.2) and (A.3). Combining the evolution equation for σ^2 with (A.4) and (A.5), one obtains an equation for the evolution for a^2 in principal coordinates:

$$\frac{da^2}{dt} = 2a^2 d_{11} + \frac{\nu}{2\sigma^2}(1 - a^4)$$

From (A.6), one arrives at an evolution equation for the orientation of the basis function. Changing back to standard coordinates, one obtains the dynamical system presented in §3. The decomposition of evolution equation for the orientation into rotational and strain-induced motions in equation (3.12) is accomplished through further algebraic manipulation.

REFERENCES

[1] J. T. Beale and A. Majda, *Vortex methods. II: Higher order accuracy in two and three dimensions*, Mathematics of Computation, 39 (1982), pp. 29–52.

[2] G.-H. Cottet and P. D. Koumoutsakos, *Vortex methods: theory and practice*, Cambridge University Press, Cambridge; New York, 2000.

[3] J. Goodman, T. Y. Hou, and J. Lowengrub, *Convergence of the point vortex method for the 2D Euler equations*, Communications on Pure and Applied Mathematics, XLIII (1990), pp. 415–430.
[4] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.
[5] B. Hancock, *White house supporting open source code*, Computers & Security, 19 (2000), pp. 577–578.
[6] P. Jones, *Open(source)ing the doors for contributor-run digital libraries*, Comm. ACM, 44 (2001), pp. 45–46.
[7] S. Kida, *Motion of an elliptic vortex in a uniform shear flow*, J. Phys. Soc. Japan, 50 (1981), pp. 3517–3520.
[8] T. Kida, *Theoretical and numerical results of a deterministic two-dimensional vortex method*, Sadhana, 23 (1998), pp. 419–441.
[9] T. Kida and T. Nakajima, *Core spreading vortex methods in two-dimensional viscous flows.*, Comput. Methods Appl. Mech. Engrg., 160 (1998), pp. 273–298.
[10] T. Kida, T. Nakajima, and H. Suemitsu, *Second order core spreading vortex method in two-dimensional viscous flows*, JSME Int. J. Series B - Fluids and Therm. Engg., 41 (1998), pp. 441–446.
[11] H. Lamb, *Hydrodynamics*, Cambridge University Press, 1993, ch. IV, VII, pp. 84–86, 232–233.
[12] A. Leonard, *Vortex methods for flow simulation*, J. Comp. Phys., 37 (1980), pp. 289–335.
[13] A. Leonard, *Aiaa 97-0204: Large-eddy simulation of chaotic convection and beyond*, in 35th Aerospace Sciences Meeting & Exhibit, American Institute of Aeronautics and Astronautics, 1997.
[14] J. S. Lowengrub and M. J. Shelley, *Smooth grid methods for the vorticity formulation of the Euler equations*, in Lectures in applied mathematics, vol. 28, American Mathematical Society, 1991, pp. 423–432.
[15] J. S. Lowengrub, M. J. Shelley, and B. Merriman, *High-order and efficient methods for the vorticity formulation of the Euler equations.*, SIAM J. Sci. Comp., 14 (1993), pp. 1107–1142.
[16] Z. Y. Lu and T. J. Ross, *Diffusing-vortex numerical scheme for solving incompressible Navier-Stokes equations*, J. Comp. Phys., 95 (1991), pp. 400–435.
[17] J. S. Marshall and J. R. Grant, *A method for determining the velocity induced by highly anisotropic vorticity blobs*, J. Comp. Phys., 126 (1996), pp. 286–298.
[18] E. Meiburg, *Incorporation and test of diffusion and strain effects in the two-dimensional vortex blob technique*, J. Comp. Phys., 82 (1989), pp. 85–93.
[19] A. Ojima and K. Kamemoto, *Numerical simulation of unsteady flow around three dimensional bluff bodies by an advanced vortex method.*, JSME Int. J. Series B - Fluids and Therm. Mixing, 43 (2000), pp. 127–135.
[20] E. S. Raymond, *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*, O'Reilly & Associates, Inc., Sebastipol CA, 1999.
[21] L. Rosenhead, *The point vortex approximation of a vortex sheet*, Proceedings of the Royal Society of London Series A, 134 (1932), pp. 170–192.
[22] L. F. Rossi, *Resurrecting core spreading methods: A new scheme that is both deterministic and convergent*, SIAM J. Sci. Comp., 17 (1996), pp. 370–397.
[23] ———, *Merging computational elements in Lagrangian simulations*, SIAM J. Sci. Comp., 18 (1997), pp. 1014–1027.
[24] L. F. Rossi, J. F. Lingevitch, and A. J. Bernoff, *Quasi-steady monopole and tripole attractors for relaxing vortices*, Physics of Fluids, 9 (1997), pp. 2329–2339.
[25] J. A. Sethian, *A brief overview of vortex methods*, in Vortex Methods and Vortex Motion, K. Gustafson and J. A. Sethian, eds., SIAM Frontiers in Applied Mathematics, 1990.
[26] D. Shiels, *Simulation of controlled bluff body flow with a viscous vortex method*, PhD thesis, California Institute of Technology, Pasadena, CA, 1998.
[27] Z.-H. Teng, *Elliptic-vortex method for incompressible flow at high Reynolds number.*, J. Comp. Phys., 46 (1982), pp. 54–68.
[28] ———, *Variable-elliptical-vortex method for incompressible flow simulation.*, J. Comp. Math., 4 (1986), pp. 255–262.
[29] ———, *Convergence of the variable-elliptic-vortex method for Euler equations*, SIAM J. Num. Anal., 32 (1995), pp. 754–774.
[30] H. Wang and C. Wang, *Open source software adoption: a status report*, IEEE Software, (2001), pp. 90–95.
[31] G. S. Winckelmans and A. Leonard, *Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows.*, Journal of Computational

Physics, 109 (1993), pp. 247–273.