EVALUATION OF THE BIOT-SAVART INTEGRAL FOR DEFORMABLE ELLIPTICAL GAUSSIAN VORTEX ELEMENTS [*]

LOUIS F. ROSSI[†]

Abstract.
This paper introduces two techniques for approximating the Biot-Savart integral for deforming elliptical Gaussian functions. The primary motivation is to develop a high spatial accuracy vortex method. The first technique is a regular perturbation of the streamfunction in the small parameter $\epsilon = \frac{a-1}{a+1}$ where a^2 is the aspect ratio of the basis function. This perturbative technique is suitable for direct interactions. In the far field, the paper studies the applicability of the fast multipole method for deforming elliptical Gaussians since the multipole series are divergent. The noncompact basis functions introduce a new computational length scale that limits the efficiency of the multipole algorithm but by imposing a lower bound on the finest mesh size, one can approximate the far-field streamfunction to any specified tolerance.

Key words. computational fluid dynamics, vortex methods, deforming elements, elliptical Gaussians, fast summation

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

1. Introduction. In this paper, we explore the calculation of Biot-Savart integrals for elliptical Gaussian basis functions,

$$\psi = -\frac{1}{4\pi} \iint_{-\infty}^{\infty} \log(|\vec{x} - \vec{s}|^2)\phi(\vec{s})d\vec{s}, \tag{1.1a}$$

$$\phi(\vec{s}; \sigma, a) = \frac{1}{4\pi\sigma^2} \exp\left[\frac{-(x^2/a^2 + y^2a^2)}{4\sigma^2}\right], \tag{1.1b}$$

to determine the flow field and its derivatives induced by an elliptical Gaussian distribution of vorticity, or blob, in a two-dimensional incompressible fluid. The core size σ^2 and the aspect ratio a^2 are included as parameters. Translations and rotations of ϕ can be introduced through the symmetries in (1.1a), but we exclude them now for simplicity. There are two reasons for creating a robust and accurate approximation for the Biot-Savart integral of an elliptical Gaussian. The first reason is the impetus for this research, the development of high spatial order vortex methods for two-dimensional, incompressible, viscous flows. For a vortex method, it is necessary to have a consistent description for the velocity field for it to be convergent. Recent results demonstrating that deforming elliptical Gaussian blobs can be used to achieve fourth order spatial accuracy for the convection and diffusion of passive scalars open the door to high order viscous vortex methods [29, 30]. While not emphasized in this paper, the second reason for developing methods of approximating the Biot-Savart integral is that elliptical Gaussian vortices may be useful as modeling elements. That is, one could use a small number of elliptical Gaussians to model a more complex geophysical or laboratory system. To assess the interactions within the reduced system, one would need a means of evaluating the Biot-Savart integral of each element. These reduced models yield helpful insights without resorting to specialized, large-scale simulations. For example, Knio, Collorec and Juvé studied sound emissions from

[*] Portions of this work was supported by NSF grant DMS-9971800. Computations supported by NSF SCREMS DMS-0322583.

[†] Copyright 2005 Louis F. Rossi. Department of Mathematical Sciences, University of Delaware. Newark, DE 19716, USA (rossi@math.udel.edu)

regular and chaotic systems of discrete vortices using three Gaussian vortices [17]. To understand the vortex core deformations, they refined their computation by using a large number of Gaussian elements to approximate the initial larger Gaussian. Deforming elliptical Gaussians offer more internal degrees of freedom for reduced models like these. Another use for exact solutions are as base states for stability studies. For instance, Sipp et. al. recently studied the stability of a Gaussian vortex in an inhomogeneous fluid [36]. Similarly, a number of investigations have been conducted using interacting elliptical patches of constant vorticity (Kida vortices) to understand vortex mergers [25, 33, 44]. Elliptical Gaussian in linear flow field provides another family of base states for investigators to explore.

The primary purpose of this work is to the development of high accuracy viscous vortex methods. Vortex methods are a category of Lagrangian schemes long associated with successful computation of unsteady flow fields. For example, Rosenhead used eight point vortices to compute the dynamics of an inviscid mixing layer in 1931 [26]. The common ingredient in all vortex methods is that the vorticity field of the flow is approximated by a linear combination of localized basis functions. These basis functions move with the local flow field which must determined via a Biot-Savart integral over the basis functions. Overviews of a variety of vortex method techniques and applications can be found in [6, 19, 32] although new methods and techniques emerge regularly. Some more recent developments include large scale engineering applications high Reynolds number flow past bluff bodies by Ploumhans et. al. [24] and the use of radial basis functions for remeshing by Barba et. al [4]. This paper focuses on deforming elliptical Gaussian basis functions because they provide a means of achieving high spatial accuracy for fully viscous flow. The use of deforming blobs for the 2D, incompressible Navier-Stokes equations

$$\omega_t + \vec{u}\cdot\nabla\omega = \frac{1}{\mathrm{Re}}\nabla^2\omega, \tag{1.2a}$$

$$\nabla\cdot\vec{u} = 0, \tag{1.2b}$$

(where $\vec{u}$ is the fluid velocity, $\omega = (\nabla\times\vec{u})\cdot\widehat{z}$ is the fluid vorticity and Re is the Reynolds number) is a natural extension of existing work both on core spreading techniques for Navier-Stokes and on the use of anisotropic elements to facilitate greater adaptivity and accuracy for Euler equations.

Core spreading has considerable appeal because it is deterministic, grid-free and exact for uniform flow fields. Furthermore, the uncorrected algorithm is easy to implement and computationally efficient. To the knowledge of the author, core spreading was first proposed by Leonard [19] but concerns quickly arose over the consistency of this approach [12] because the simplest implementation of the scheme imposes a lower bound on the core size of the computational elements. The method was corrected in [27] with a consistent splitting algorithm and augmented with a merging algorithm [28]. Others have built upon these topics [15, 22, 34, 35]. Also, there have been engineering computations with *uncorrected* core spreading methods (for example [16, 40, 41, 42]) but the accuracy of such methods is questionable in lieu of the finite core size issue as well as the lack of consistency in the core spreading formulation. Barba explores the numerical issues involved in [3]. While there is cause for concern about a lack of communication across disciplines, corrected core spreading is a strong and robust method for unsteady flows.

Still, core spreading is one of a small number of families of schemes featuring deforming blobs. There have been numerous efforts to use anisotropic elements for

Euler and Navier-Stokes equations as a means of improving the efficiency of computations in large aspect ratio flows, a common situation in flows near solid boundaries. Teng used rigid and deforming elliptical patches to resolve boundary layers and vortex sheets more efficiently [37, 38, 39]. He establishes a theoretical $O(l^2 \log l)$ rate of convergence for his method where l is the core width of the computational elements. Marshall and Grant designed highly anisotropic elements to satisfy the no-slip, no normal flow boundary conditions [21]. Ojima and Kamemoto developed a scheme for large scale simulations using deforming vortex 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 to avoid having to calculate the Biot-Savart integral for anisotropic elements [23]. Rigorous convergence studies have not been reported for this method. Leonard used deforming elliptical Gaussian basis functions to simulate the convection *and diffusion* of passive scalar quantities, (1.2a) but where $\vec{u}$ is specified, motivated by the fact that they remain self-similar under the linearized convection-diffusion equations [20]. Also, Moeleker and Leonard propose using anisotropic elements with velocity field corrections based on subgrid scale for when computing Gaussian filtered scalar quantities though they report modest gains in accuracy at a high computational cost [22].

This paper builds upon findings that fourth order spatial accuracy can be achieved with deforming elements when the basis functions advect with a velocity field given by the physical velocity field plus a velocity curvature correction [29]. These studies with an uncorrected method are limited to convection-diffusion equations. However, the analysis suggests that the same performance should be achieved for Navier-Stokes as long as the velocity field computed via the Biot-Savart integral and the exact field are close to one another because the errors are introduced through the nonlinear term at second order. In §4, we shall see that this is the case. For convection-diffusion equations, the advantages of using more accurate but more computationally complex basis functions over less accurate simpler radial basis functions are realized at moderate problem sizes [30]. In short, deforming basis functions are better than axisymmetric basis functions in the sense that the former can compute a more accurate solution in the same amount of CPU time. This paper is a step in the same direction, building a better vortex method for Navier-Stokes. The vortex method described in this paper is an elliptical corrected core spreading vortex method and for convenience, it shall be termed ECCSVM. It is beyond the scope of this paper to explore corrections (the first "C") but techniques such as remeshing, splitting and merging are applicable to deformable blobs as well as axisymmetric blobs.

For a vortex method to be viable for scientific computation, it is not enough to evaluate the Biot-Savart integral directly to find the velocity field at each basis function location. If this were the case, the computational complexity of determining the velocity field at N blob positions would be $O(N^2)$. A number of fast summation algorithms have been developed over the past two decades that are capable of reducing certain particle computations from $O(N^2)$ to $O(N \log N)$ or even $O(N)$. These methods all hinge upon a separation of length scales. For instance, the Greengard-Rohklin algorithm uses Laurent expansions of complex Coulomb potentials to collect far field influences. The effects of distant elements are approximated as an aggregate while velocity contributions from nearby elements are computed directly. The Greengard-Rohklin fast multipole method (FMM) has demonstrably reduced the computational complexity from $O(N^2)$ to $O(N)$ for potential kernels [13]. The multipole coefficients for the Laurent series can be collected on an hierarchy of meshes over the domain, so that the separation of near and far fields can be nested. There have been many extensions and augmentations to FMM (for instance [1, 2, 5, 14, 43]). The original FMM concepts have also been extended for fast polynomial interpolation and differentiation [10] and for a much broader class of velocity kernels [11]. An alternative to this algorithm based on Taylor series expansions has been analyzed and demonstrated by Draghicescu and Draghicescu [7, 8, 9]. This technique requires that one be able to Taylor expand the core shape function, and it achieves $O(N \log N)$ complexity. While the Draghicescu algorithm offers more flexibility in the types of core shapes one could use, the method requires that all elements have the same shape, so the Draghicescu method would not be suitable for the evolving anisotropic elements that are the focus of this paper. One of several difficulties in using FMM under these circumstances is that the Biot-Savart integral (i.e. the velocity kernel) cannot be written in terms of elementary functions. The approach developed in this paper is to expand the Coulomb potential inside the integral (1.1a) as a Laurent series, and then perform the integration term by term. We shall see later that the resulting series is divergent but useful as an approximation, and will yield an accurate $O(N)$ algorithm for finding the velocity field induced by a distant collection of elliptical Gaussian basis functions.

This paper has a simple organizational structure. This section reviews different perspectives and contributions in the use of anisotropic elements for particle computations. The next section §2 derives and demonstrates how one can approximate the streamfunction and its derivatives for an elliptical Gaussian basis function for direct pairwise interactions. In §3, the fast multipole method is modified for use with anisotropic elliptical Gaussians. In this section, we rely heavily on the work of Greengard and Rohklin [13]. Finally in §4, we certify these methods by approximating solutions to the Navier-Stokes for the Lamb-Oseen vortex and compare the numerical solutions with an exact solution.

2. Asymptotic evaluations for direct interactions in the near field. The evolution equations for the elliptical Gaussian basis functions (1.1b) depend upon $\vec{u}$ and $D\vec{u}$ [1], but unlike the most other basis functions used for vortex methods, there is no known elementary formula for these induced fields. However, the induced velocity and velocity derivatives can be written as a regular perturbation in the small parameter

$$\epsilon = \frac{a-1}{a+1}. \tag{2.1}$$

For simplicity, we let $R^2 = x^2/a^2 + y^2a^2$, so that (1.1b) becomes,

$$\phi(R;\sigma,a) = \frac{1}{4\pi\sigma^2} e^{-R^2/4\sigma^2}. \tag{2.2}$$

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

[1] See [29] for the dynamics, and fourth order accuracy also requires knowing the second partial derivatives of the velocity components.

axes of l_1 and l_2 can be determined using elliptical coordinates [18]:

$$\psi = \begin{cases} \frac{1}{2\pi(l_1+l_2)}\left(\frac{x^2}{l_1}+\frac{y^2}{l_2}\right), & (x,y)\in\mathcal{E}(l_1,l_2) \\ \frac{1}{2\pi}\left[\ln\left(\frac{\alpha+\beta}{l_1+l_2}\right)+\frac{\frac{x^2}{\alpha}+\frac{y^2}{\beta}}{\alpha+\beta}\right], & (x,y)\notin\mathcal{E}(l_1,l_2) \end{cases} \tag{2.3a}$$

$$\alpha = \sqrt{l_1^2+\xi}, \tag{2.3b}$$

$$\beta = \sqrt{l_2^2+\xi}, \tag{2.3c}$$

$$1 = \left(\frac{x^2}{l_1^2+\xi}\right)+\left(\frac{y^2}{l_2^2+\xi}\right), \tag{2.3d}$$

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 (2.2), the constituent patch would have density $\pi R^2(-\partial_R\phi)dR$ 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 (x_*, y_*):

$$\psi(x_*,y_*) = -\int_0^{R_*}\psi_1\partial_R\phi R^2 dR - \int_{R_*}^{\infty}\psi_2\partial_R\phi dR, \tag{2.4a}$$

$$\psi_1 = \frac{1}{2}\left[\frac{\left(\frac{{x_*}^2}{\alpha}+\frac{{y_*}^2}{\beta}\right)}{\alpha+\beta}+\ln\left(\frac{\alpha+\beta}{Ra+R/a}\right)\right], \tag{2.4b}$$

$$\psi_2 = \frac{1}{2}\frac{\frac{{x_*}^2}{a}+{y_*}^2 a}{a+1/a}, \tag{2.4c}$$

$$\alpha = \sqrt{R^2a^2+\xi}, \tag{2.4d}$$

$$\beta = \sqrt{\frac{R^2}{a^2}+\xi}. \tag{2.4e}$$

Rather than using the implicit relation in equation (2.3d), we can use the quadratic solution

$$\xi = \frac{1}{2}\left\{{\rho_*}^2 - R^2\left(a^2+\frac{1}{a^2}\right)+\sqrt{\left[R^2\left(a^2+\frac{1}{a^2}\right)-{\rho_*}^2\right]^2+4R^2({R_*}^2-R^2)}\right\}, \tag{2.5}$$

where ${\rho_*}^2 = {x_*}^2+{y_*}^2$ and ${R_*}^2 = \frac{{x_*}^2}{a^2}+{y_*}^2a^2$. Important limits to be used later are $\xi \to {\rho_*}^2$ as $R \to 0$ and $\xi \to 0$ as $R \to R_*$.

The second term on the right side of (2.4a) is elementary to integrate because ψ_2 does not vary with R. ψ_1 is not readily expressed in terms of the variable R making the first term problematic. However, one can expand ψ_1 in powers of the small parameter ϵ. Expanding in ϵ rather than R has the advantage that unlike R/σ, ϵ is a small number that is less than 1. The variable R is not guaranteed to be small relative to σ since one may wish to use the streamfunction anywhere in the domain. Thus, 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, R_* is a parameter representing one possible combination of x_* and y_*. One possible complement is T_* where ${T_*}^2 = {x_*}^2/a^2 - {y_*}^2a^2$. Combined with the fact that the streamfunction must retain symmetries about the major and minor axes, the streamfunction can be expressed as a function of R_* and T_*. One could use radially symmetric coordinates such as ${x_*}^2 + {y_*}^2$ and ${x_*}^2 - {y_*}^2$, but it was found that these parameters yield a less accurate streamfunction approximation.

To perform an expansion of ψ_1, we make the substitutions,

$$
{x_*}^2 = \frac{a^2}{2}\left({R_*}^2 + {T_*}^2\right), \tag{2.6a}
$$

$$
{y_*}^2 = \frac{1}{2a^2}\left({R_*}^2 - {T_*}^2\right), \tag{2.6b}
$$

$$
a = \frac{1+\epsilon}{1-\epsilon}, \tag{2.6c}
$$

the latter arising from (2.1). Thus, it follows that

$$
{\rho_*}^2 = {R_*}^2 + 4\epsilon{T_*}^2 + 8\epsilon^2{R_*}^2 + 12\epsilon^3{T_*}^2 + 16\epsilon^4{R_*}^4 + O(\epsilon^5). \tag{2.7}
$$

Substituting this expression into the streamfunctions (2.4e) and (2.5), one can obtain an asymptotic expression for the streamfunction of the elliptical patch. If one expands these expressions out to a power of ϵ^n, then one obtains a polynomial in R of order $2n$. For example, if we expand in powers of ϵ for $n = 4$, we capture the following approximation for the streamfunction:

$$
\begin{aligned}
\psi_1 =& \frac{1}{4}\left[1 + \ln\left(\frac{{R_*}^2}{R^2}\right)\right] + \epsilon\left(\frac{{T_*}^2}{{R_*}^2} - \frac{1}{2}\frac{{T_*}^2R^2}{{R_*}^4}\right) + \\
& \epsilon^2\left[1 - 2\frac{{T_*}^4}{{R_*}^4} + \left(-2 + 4\frac{{T_*}^4}{{R_*}^4}\right)\frac{R^2}{{R_*}^2} + \left(1 - 2\frac{{T_*}^4}{{R_*}^4}\right)\frac{R^4}{{R_*}^4}\right] + \\
& \epsilon^3\left[\frac{16}{3}\frac{{T_*}^6}{{R_*}^6} - 5\frac{{T_*}^2}{{R_*}^2} + \left(\frac{37}{2}\frac{{T_*}^2}{{R_*}^2} - 24\frac{{T_*}^6}{{R_*}^6}\right)\frac{R^2}{{R_*}^2} + \left(-24\frac{{T_*}^2}{{R_*}^2} + 32\frac{{T_*}^6}{{R_*}^6}\right)\frac{R^4}{{R_*}^4} + \right. \\
& \left.\left(10\frac{{T_*}^2}{{R_*}^2} - \frac{40}{3}\frac{{T_*}^6}{{R_*}^6}\right)\frac{R^6}{{R_*}^6}\right] + \\
& \epsilon^4\left[-4 + 20\frac{{T_*}^4}{{R_*}^4} - 16\frac{{T_*}^8}{{R_*}^8} + \left(20 - 136\frac{{T_*}^4}{{R_*}^4} + 128\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^2}{{R_*}^2} + \right. \\
& \left(-42 + 324\frac{{T_*}^4}{{R_*}^4} - 320\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^4}{{R_*}^4} + \left(40 - 320\frac{{T_*}^4}{{R_*}^4} + 320\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^6}{{R_*}^6} + \\
& \left.\left(-14 + 112\frac{{T_*}^4}{{R_*}^4} - 112\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^8}{{R_*}^8}\right] + O(\epsilon^5)
\end{aligned} \tag{2.8}
$$

$$
\psi_2 = \frac{1}{2}\left[\frac{1}{2}{R_*}^2 + \left(\epsilon - \epsilon^3\right){T_*}^2\right] + O(\epsilon^5). \tag{2.9}
$$

Using this expansion, the calculation of an approximate streamfunction together with its derivatives reduces to computing moments of elliptical Gaussians. Since ψ_2 does not vary with R, the second term of (2.4a) can be evaluated with the exact form of

ψ_2 from (2.4e) or with the approximate form from (2.9). The advantage to using the approximation is that it makes the integrand and one derivative of the integrand continuous across R_*, so that one need not use Leibnitz's rule when calculating velocities, but this is a minor consideration.

Next, we would like to calculate the derivatives of ψ. To be more precise, knowing the n^{th} derivatives of ψ, we would like to be able to quickly compute the $(n+1)^{\text{st}}$ derivatives. While these operations can be performed using finite differences, the cost is high because ψ would have be evaluated many times over a stencil near the origin. Instead, we shall determine the derivatives exactly from the asymptotic expression.

The regular perturbation in ϵ approximates ψ_1 as a polynomial in R whose coefficients depend upon ratios of powers T_* and R_*. Using the polynomial approximation, finding ψ in (2.4a) reduces to summing moments of the Gaussian. Differentiation of ψ reduces to a transformation of the polynomial coefficients in (2.8) followed by the summation of all the resulting moments. For instance, we can rewrite (2.8) as

$$\begin{aligned}
\psi_1 =& \frac{1}{4}\left[1+\ln\left(\frac{{R_*}^2}{R^2}\right)\right]+\\
&\left(\epsilon^2-4\epsilon^4\right)+\left(\epsilon-5\epsilon^3\right)\frac{{T_*}^2}{{R_*}^2}+\left(-2\epsilon^2+20\epsilon^4\right)\frac{{T_*}^4}{{R_*}^4}+\frac{16}{3}\epsilon^3\frac{{T_*}^6}{{R_*}^6}-16\epsilon^4\frac{{T_*}^8}{{R_*}^8}+\\
&\left[\left(-2\epsilon^2+20\epsilon^4\right)+\left(-\frac{1}{2}\epsilon+\frac{37}{2}\epsilon^3\right)\frac{{T_*}^2}{{R_*}^2}\right.\\
&\left.+\left(4\epsilon^2-136\epsilon^4\right)\frac{{T_*}^4}{{R_*}^4}-24\epsilon^3\frac{{T_*}^6}{{R_*}^6}+128\epsilon^4\frac{{T_*}^8}{{R_*}^8}\right]\frac{R^2}{{R_*}^2}+\\
&\left[\left(\epsilon^2-42\epsilon^4\right)-24\epsilon^3\frac{{T_*}^2}{{R_*}^2}+\left(-2\epsilon^2+324\epsilon^4\right)\frac{{T_*}^4}{{R_*}^4}+32\epsilon^3\frac{{T_*}^6}{{R_*}^6}-320\epsilon^4\frac{{T_*}^8}{{R_*}^8}\right]\frac{R^4}{{R_*}^4}+\\
&\left(40\epsilon^4+10\epsilon^3\frac{{T_*}^2}{{R_*}^2}-320\epsilon^4\frac{{T_*}^4}{{R_*}^4}-\frac{40}{3}\epsilon^3\frac{{T_*}^6}{{R_*}^6}+320\epsilon^4\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^6}{{R_*}^6}+\\
&\left(-14\epsilon^4+112\epsilon^4\frac{{T_*}^4}{{R_*}^4}-112\epsilon^4\frac{{T_*}^8}{{R_*}^8}\right)\frac{R^8}{{R_*}^8}+O(\epsilon^5).
\end{aligned} \tag{2.10}$$

If we terminate the expansion of ψ_1 after finding the coefficients for ϵ^n, the result will be a polynomial of order $2n$ in R.

Differentiating the approximate streamfunction with respect to position, x_* or y_* reduces to differentiating ratios of the form $\frac{{T_*}^{2m}}{{R_*}^{2n}}$ where $m \leq n$. For instance, if $m > 0$,

$$\frac{\partial}{\partial x_*}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right)=\left(m\frac{{T_*}^{2(m-1)}}{{R_*}^{2n}}-n\frac{{T_*}^{2m}}{{R_*}^{2(n+1)}}\right)\frac{2x_*}{a^2}, \tag{2.11a}$$

$$\frac{\partial}{\partial y_*}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right)=-\left(m\frac{{T_*}^{2(m-1)}}{{R_*}^{2n}}+n\frac{{T_*}^{2m}}{{R_*}^{2(n+1)}}\right)2y_*a^2. \tag{2.11b}$$

Similarly, when $m > 0$, we find

$$\frac{\partial^2}{\partial {x_*}^2}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right) = \left(m\frac{{T_*}^{2(m-1)}}{{R_*}^{2n}} - n\frac{{T_*}^{2m}}{{R_*}^{2(n+1)}}\right)\frac{2}{a^2} + \left[m(m-1)\frac{{T_*}^{2(m-2)}}{{R_*}^{2n}} - 2mn\frac{{T_*}^{2(m-1)}}{{R_*}^{2(n+1)}} + n(n+1)\frac{{T_*}^{2m}}{{R_*}^{2(n+2)}}\right]\frac{4{x_*}^2}{a^4}. \quad (2.12)$$

However, if we make the substitution $4{x_*}^2/a^4 = 2({R_*}^2 + {T_*}^2)/a^2$, and perform similar substitutions on terms involving ${y_*}^2 = ({R_*}^2 - {T_*}^2)/(2a^2)$, we determine the following relations:

$$\frac{\partial^2}{\partial {x_*}^2}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right) = \frac{2}{a^2}\left[m(m-1)\frac{{T_*}^{2(m-2)}}{{R_*}^{2(n+1)}} + m(m-2n)\frac{{T_*}^{2(m-1)}}{{R_*}^{2n}} + n(n-2m)\frac{{T_*}^{2m}}{{R_*}^{2(n+1)}} + n(n+1)\frac{{T_*}^{2(m+1)}}{{R_*}^{2(n+2)}}\right], \quad (2.13a)$$

$$\frac{\partial^2}{\partial {y_*}^2}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right) = 2a^2\left[m(m-1)\frac{{T_*}^{2(m-2)}}{{R_*}^{2(n+1)}} + m(2n-m)\frac{{T_*}^{2(m-1)}}{{R_*}^{2n}} + n(n-2m)\frac{{T_*}^{2m}}{{R_*}^{2(n+1)}} - n(n+1)\frac{{T_*}^{2(m+1)}}{{R_*}^{2(n+2)}}\right], \quad (2.13b)$$

$$\frac{\partial^2}{\partial x_* \partial y_*}\left(\frac{{T_*}^{2m}}{{R_*}^{2n}}\right) = 4x_*y_*\left[m(1-m)\frac{{T_*}^{2(m-2)}}{{R_*}^{2n}} + n(n+1)\frac{{T_*}^{2m}}{{R_*}^{2(n+2)}}\right]. \quad (2.13c)$$

Relations (2.11) and (2.13) create an opportunity to calculate the partial derivatives of ψ. If we represent the streamfunction as a vector of coefficients with indices m and p because $n = m + p$ for terms of the form $\frac{{T_*}^m}{{R_*}^n}R^{2p}$ as shown in (2.10), the coefficients of the partial derivatives of ψ_1 are linear combinations of the coefficients for ψ_1. Specific algorithms arising from these expressions provided in Appendix A.

Now we can approximate the Biot-Savart integral by calculating moments of the elliptical Gaussian. For instance, if $n = 4$, we calculate the elementary integral,

$$\begin{aligned}
I =& -\int_0^{R_*} R^2(C_0 + C_1R^2 + C_2R^4 + C_3R^6 + C_4R^8)\partial_R\phi dR - C_5\phi(R_*) \\
=& \frac{\gamma}{2\pi}\left\{2C_0 + 16C_1\sigma^2 + 192C_2\sigma^4 + 3072C_3\sigma^6 + 61440C_4\sigma^8\right. \\
& - \left[C_0\left(\frac{{R_*}^2}{2\sigma^2} + 2\right) + C_1\left(\frac{{R_*}^4}{2\sigma^2} + 4{R_*}^2 + 16\sigma^2\right)\right. \\
& + C_2\left(\frac{{R_*}^6}{2\sigma^2} + 6{R_*}^4 + 48{R_*}^2\sigma^2 + 192\sigma^4\right) \\
& + C_3\left(\frac{{R_*}^8}{2\sigma^2} + 8{R_*}^6 + 96\sigma^2{R_*}^4 + 768\sigma^4{R_*}^2 + 3072\sigma^6\right) \\
& + C_4\left(\frac{{R_*}^{10}}{2\sigma^2} + 10{R_*}^8 + 160\sigma^2{R_*}^6 + 1920\sigma^4{R_*}^4 + 15360\sigma^6{R_*}^2 + 61440\sigma^8\right) \\
& \left.\left. - \frac{C_5}{2\sigma^2}\right]\exp\left(-\frac{{R_*}^2}{4\sigma^2}\right)\right\}.
\end{aligned} \quad (2.14)$$

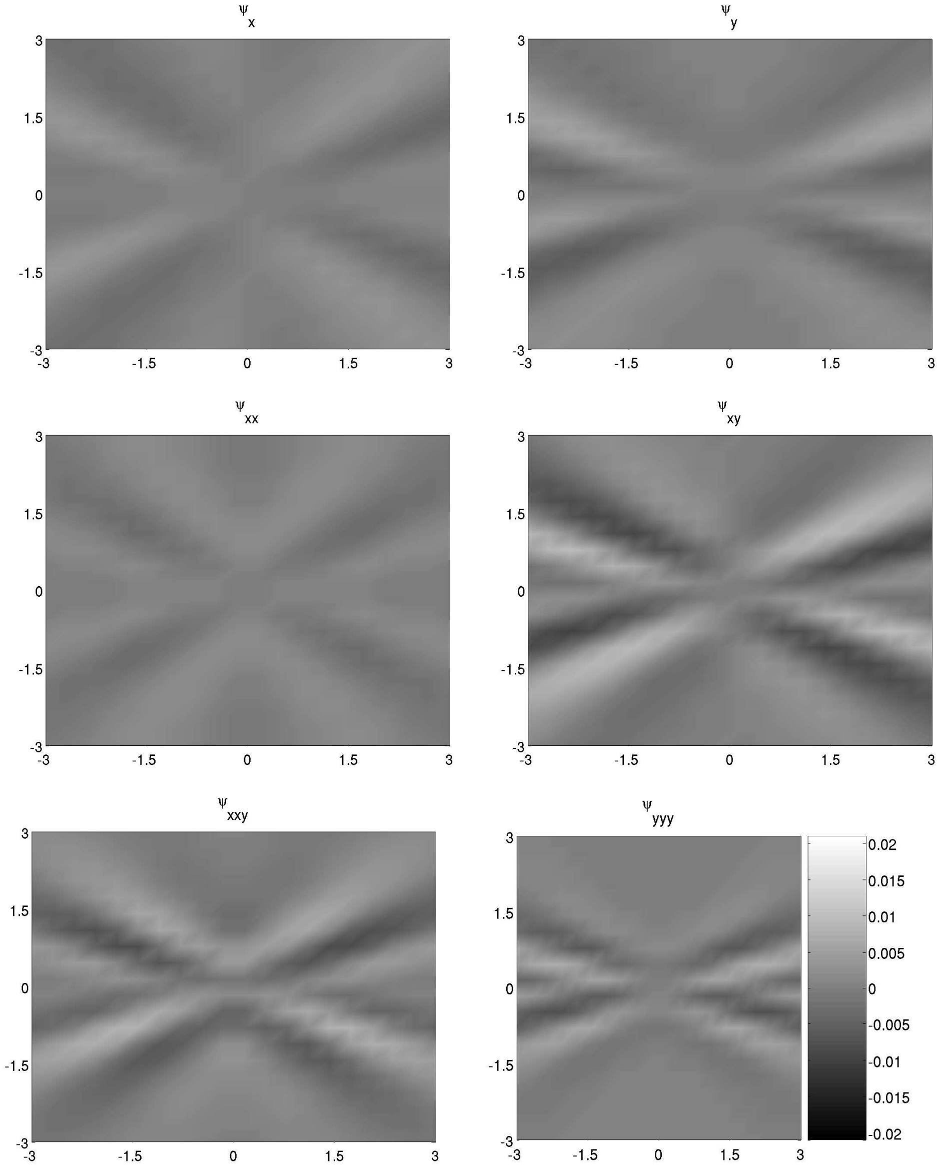

FIG. 2.1. *Relative field errors for the fourth order approximation of ψ for $\epsilon = 0.20$ (aspect ratio 2.25) and $\sigma = 1/4$. The relative error is calculated using (2.15) where the reference solution is calculated using a high precision quadrature. The scale as shown in the colorbar on the ψ_{yyy} plot spanning an relative error of $\pm 2\%$ is used for all the plots.*

where I could approximate ψ_x, ψ_y, ψ_{xx}, etc, and the C_i are polynomials in $\frac{T_*{}^m}{R_*{}^n}$.

To assess the accuracy of these approximations, the errors associated with the approximate solutions are compared to a reference solution using finite differences on a precise double-exponential quadrature of (2.4e) with fourteen digits of precision so that the third order finite differences could be adequately resolved. The results of a

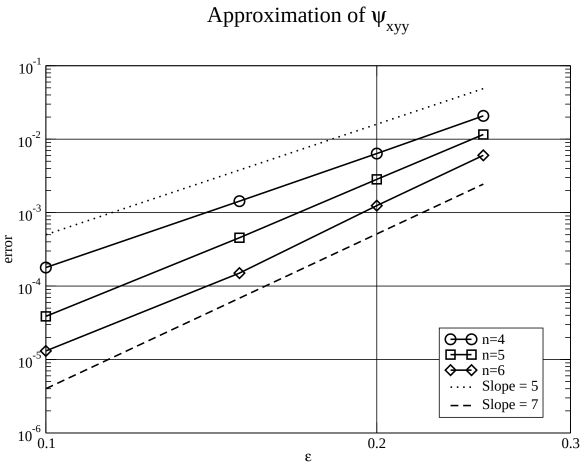

FIG. 2.2. *Behavior of asymptotic approximation for the streamfunction and its derivatives as one varies ϵ and n. The graph presents the absolute error versus ϵ on logarithmic axes along with reference slopes to show that the predicted convergence rates are realized.*

typical calculation is shown in Fig. 2.1. The displayed relative error e is

$$e = \frac{\|f(x,y) - f_{\text{ref}}(x,y)\|_{\text{sup}}}{\|f_{\text{ref}}\|_{\text{sup}}} \tag{2.15}$$

where $f(x,y)$ is the approximation, $f_{\text{ref}}(x,y)$ is the reference solution and the supremum is taken over the displayed domain.

As computational elements grow more eccentric and ϵ grows closer to one, the approximation becomes less accurate. On the other hand, the regular perturbation can be expanded for any n. Thus, there are competing trends. Fig. 2.2 is an exploration of variations in ϵ and n, and we see the predicted rates of convergences are realized in practice.

3. Far field calculations using fast multipoles. Far field calculations of anisotropic elements can be handled separately and more efficiently than direct calculations using fast summation. Fast summation algorithms all hinge upon a separation of length scales, either artificial or physical. For instance, the Greengard-Rohklin algorithm uses Laurent expansions for complex potential to collect far field influences where the series is guaranteed to converge. At first glance, the Greengard-Rokhlin would also appear to be unsuitable for this method because the basis functions have infinite support and the kernel is singular, so the Laurent series will not converge at any point in the domain. We shall see in this section that lacking compact support does not completely sink the idea because the partial sums contain useful information. By necessity, some details of Greengard-Rokhlin summation will be presented, but other aspects not related to anisotropic elements will be omitted in the interest of brevity. For a complete discussion of this method, the reader is referred to Greengard and Rokhlin's original article [13] or its successors some of which are cited in §1.

To begin, we will express positions in the domain (x,y) as a complex variable $z = x+iy$. Then, the complex potential for a point vortex at the origin with circulation γ will be

$$\varphi(z) = -\frac{i\gamma}{2\pi}\ln z. \tag{3.1}$$

If the velocity field has horizontal and vertical components u and v respectively, then $\varphi' = u - iv$. Velocity derivatives in the far field can be computed by repeated differentiation of φ. If the point vortex were at a point z_*, the complex potential could still be evaluated relative to the origin using a power series expansion.

$$\begin{aligned}\varphi(z) &= -\frac{i\gamma}{2\pi}\ln(z - z_*)\\ &= -\frac{i\gamma}{2\pi}\left[\ln z + \ln\left(1 - \frac{z_*}{z}\right)\right]\\ &= -\frac{i\gamma}{2\pi}\left[\ln z - \sum_{j=1}^{\infty}\frac{1}{j}\left(\frac{z_*}{z}\right)^j\right] \end{aligned} \tag{3.2}$$

Thus, a singular potential displaced from the origin can be represented through a multipole expansion relative to the origin. The power series for $\ln(1-z)$ has a radius of convergence of 1, so that as long as the point of evaluation is further from the origin than the point vortex, the series will converge. In fact, one could design a simple algorithm where one uses direct velocity calculations if, say, $|z_*/z| > 1/2$ and use the series approximation otherwise. In the latter case, one could compute enough terms of the series to satisfy any desired error tolerance. The precise ratio 1/2 is not so important, but it is crucial that one define the far field. For the remainder of this paper, we shall fix a parameter $\delta_1 < 1$, and define what is far from the origin as $|z_*/z| \leq \delta_1$. Contributions to the velocity field from elements that are not deemed far from the point of measurement will be computed directly as described in §2, but for elements that are far from the point of measurement, we shall take advantage of the series solutions described in this section. When the point of measurement z is far from the point vortex z_*, series (3.2) converges.

One can *approximate* φ when $|z_*/z| \leq \delta_1$ by truncating the convergent power series,

$$\varphi(z) = -\frac{i\gamma}{2\pi}\left[\ln z - \sum_{j=1}^{p}\frac{1}{j}\left(\frac{z_*}{z}\right)^j - \frac{1}{p+1}\left(\frac{w}{z}\right)^{p+1}\right], \tag{3.3}$$

where w is a point on the line segment connecting 0 and z_* in the complex plane. If we consider p as a controllable parameter in a calculation, we can choose p large enough to guarantee that the multipole approximation is as close as one would like to the exact potential.

Next, we can think about many elements with circulation γ_k and position z_k clustered near the origin. Following the same procedure as (3.2), we can find the complex potential at some point z relative to the origin.

$$\begin{aligned}\varphi &= -\sum_k \frac{i\gamma_k}{2\pi}\ln(z - z_k),\\ &= -\sum_k \frac{i\gamma_k}{2\pi}\left[\ln z - \sum_{j=1}^{p}\frac{1}{j}\left(\frac{z_k}{z}\right)^j - \frac{1}{p+1}\left(\frac{w_k}{z}\right)^{p+1}\right],\\ &= a_0 \ln z - \sum_{j=1}^{p}\frac{a_j}{jz^j} + R_{p+1}, \end{aligned} \tag{3.4a}$$

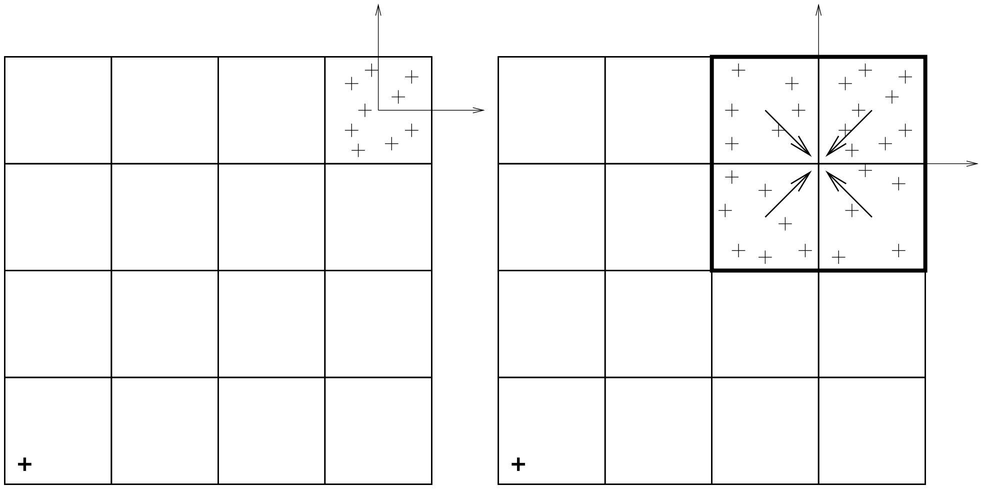

FIG. 3.1. *Overview of fast multipole summation. At left, we can see how simple cluster can speed a calculation. If we seek to calculate the streamfunction and its derivatives for the computational element (shown in bold) at the bottom left box, we can treat distant elements in the upper right box as an aggregate rather than through individual interactions. Once one calculates the multipole coefficients for all the elements in the upper right box relative to the origin labeled with axes in the figure, (3.4) can be used anywhere in the far field including the position of the basis function in the bottom left box. At right, we can see how one can save more computational effort by shifting multipole coefficients from the four small boxes in the upper right to the larger superbox by performing multipole expansion about the new origin labeled in the figure. The shifts are indicated by the bold arrows.*

where w_k is on the line segment connecting 0 and z_k in the complex plane and

$$a_0 = -\sum_k \frac{i\gamma_k}{2\pi}, \qquad a_j = -\sum_k \frac{i\gamma_k}{2\pi} z_k^j, \qquad |R_{p+1}| \le \frac{\delta_1^{p+1}}{2\pi(p+1)} \sum_k |\gamma_k|. \tag{3.4b}$$

Thus, one can calculate the multipole coefficients a_j once for a cluster of elements grouped near the origin, and use (3.4) many times in the far field as shown in Fig. 3.1.

Substantial economy is added to the technique if one shifts multipole expansions relative to one another. That is, one can expand (3.2) or (3.4a) about another point (not the origin) and combine it with other clusters. In Fig. 3.1, we see an example where four localized clusters can be combined into a single larger cluster when calculating distant interactions. The original Greengard-Rokhlin algorithm nests clusters in this arrangement to an arbitrarily fine level. The finest level can be defined as a mesh where no box has more than k elements in it, so there are no more than Nk direct interactions.

Applying multipole summation to elliptical Gaussian basis functions poses a challenge because the basis functions have infinite support, so the Laurent series will not be convergent anywhere. However, even divergent series can provide useful information. In fact, one can use the leading partial sums to compute a useful approximation for the streamfunction and its derivatives.

We begin by calculating a multipole expansion for an elliptical Gaussian basis function that is displaced from the origin to some point (x_0, y_0), with circulation γ, aspect ratio a^2, width σ and orientation θ. The complex potential would be the superposition

$$\varphi(z) = -\frac{i}{2\pi}\iint_{-\infty}^{\infty} \ln(z-z_*)\frac{\gamma}{4\pi\sigma^2}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]dx_*dy_*, \tag{3.5}$$

where $z = x + iy$, $z_* = x_* + iy_*$ and,

$$\tilde{x} = \cos(\theta)(x_* - x_0) + \sin(\theta)(y_* - y_0), \quad \tilde{y} = -\sin(\theta)(x_* - x_0) + \cos(\theta)(y_* - y_0). \tag{3.6}$$

Writing the truncated multipole expansion inside the integral, interchanging the summation and integration, and changing variables,

$$\begin{aligned}\varphi(z) =& -\frac{i\gamma}{2\pi}\iint_{-\infty}^{\infty}\left[\ln(z) - \sum_{j=1}^{p}\frac{1}{j}\left(\frac{z_*}{z}\right)^j - \frac{1}{p+1}\left(\frac{w_*}{z}\right)^{p+1}\right]\\ &\times\frac{1}{4\pi\sigma^2}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]dx_*dy_*,\\ =& -\frac{i\gamma}{2\pi}\left\{\ln z - \sum_{j=1}^{p}\frac{1}{jz^j}\iint_{-\infty}^{\infty}(z_*)^j\frac{1}{4\pi\sigma^2}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]dx_*dy_* + R_{p+1}\right\},\\ =& -\frac{i\gamma}{2\pi}\left\{\ln z - \sum_{j=1}^{p}\frac{1}{jz^j}\iint_{-\infty}^{\infty}\eta^j\frac{1}{4\pi\sigma^2}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]d\tilde{x}d\tilde{y} + R_{p+1}\right\},\end{aligned} \tag{3.7}$$

where

$$\eta = (x_0 + iy_0) + (\cos\theta + i\sin\theta)\tilde{x} + (-\sin\theta + i\cos\theta)\tilde{y} \tag{3.8}$$

and

$$R_{p+1} = -\frac{1}{4\pi\sigma^2}\iint_{-\infty}^{\infty}\frac{1}{p+1}\left(\frac{w_*}{z}\right)^{p+1}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]dx_*dy_*. \tag{3.9}$$

The R_{p+1} determines the accuracy of the truncated series, so we will compute an upper bound on this quantity. Without loss of generality, we assume that $a > 1$ and that p is odd. Then, from (3.8) and (3.9), we can apply the triangle inequality and change to polar coordinates to find that

$$\begin{aligned}|R_{p+1}| \leq& \frac{1}{p+1}\frac{1}{4\pi\sigma^2}\iint_{-\infty}^{\infty}\frac{|z_0|^{p+1}+|\tilde{z}|^{p+1}}{|z|^{p+1}}\exp\left[-\frac{\tilde{x}^2/a^2+\tilde{y}^2a^2}{4\sigma^2}\right]d\tilde{x}d\tilde{y},\\ \leq& \frac{1}{p+1}\left[\left(\frac{|z_0|}{|z|}\right)^{p+1} + \frac{2\pi}{4\pi\sigma^2}\int_0^{\infty}\frac{r^{p+2}}{|z|^{p+1}}\exp\left(-\frac{r^2}{4a^2\sigma^2}\right)dr\right],\\ \leq& \frac{1}{p+1}\left[\left(\frac{|z_0|}{|z|}\right)^{p+1} + \frac{2\pi}{4\pi\sigma^2}\int_0^{\infty}\frac{r^{p+2}}{|z|^{p+1}}\exp\left(-\frac{r^2}{4a^2\sigma^2}\right)dr\right].\end{aligned} \tag{3.10}$$

If we let

$$C_{p+2} = \frac{2\pi}{4\pi\sigma^2}\int_0^{\infty}\frac{r^{p+2}}{|z|^{p+1}}\exp\left(-\frac{r^2}{4a^2\sigma^2}\right)dr, \tag{3.11}$$

then integrating by parts, we find that

$$C_{p+2} = (p+1)\frac{2\sigma^2 a^2}{|z|^2} C_p. \tag{3.12}$$

Since $C_1 = a^2$,

$$C_{p+2} = a^2((p+1)/2)!\left(\frac{4\sigma^2 a^2}{|z|^2}\right)^{(p+1)/2}. \tag{3.13}$$

Therefore,

$$|R_{p+1}| \leq \frac{1}{p+1}\left[\left(\frac{|z_0|}{|z|}\right)^{p+1} + a^2\left(\frac{p+1}{2}\right)!\left(\frac{4\sigma^2 a^2}{|z|^2}\right)^{(p+1)/2}\right]. \tag{3.14}$$

From this expression, we see two dominant mechanisms controlling the error of the multipole series. The first term inside the []'s is the traditional ratio discussed earlier requiring that $\frac{|z_0|}{|z|} < 1$. In other words, the singular potential has no natural length scale. In the second term, we see that that core size of the anisotropic element imposes a characteristic length scale $2\sigma a$ based on the computational resolution of the basis functions. If $\frac{|z|^2}{4\sigma^2 a^2} < 1$, the upper bound on R_p in the multipole series grows without bound. However, if $\frac{|z|^2}{4\sigma^2 a^2} > 1$, the upper bound on R_p decreases initially and then grows. For fast multipole summation with point vortices, one can fix δ_1 and compute a p that makes R_{p+1} smaller than any desired tolerance, and one can set the finest mesh level for direct interactions based on memory and computational performance criteria. In other words, with point vortices, the investigator can control the number of computational elements per box at the finest mesh level and then carry out the multipole expansion as far as needed to achieve a specified tolerance. However, for elements with finite cores, this is not the case because the number of terms in the multipole series that are guaranteed to be accurate depends upon the distance to the point of measurement $|z|$ relative to the numerical length scale $2\sigma a$. So, we must define a new parameter δ_2 that characterizes the far field as $\frac{|z|^2}{4\sigma^2 a^2} > \delta_2$. One can estimate the number of useful terms by finding the value of p for which the ratio of consecutive upper bounds (3.14) is approximately one in the special case when $z_0 = 0$. This turnaround point is when

$$\begin{aligned}\left(\frac{p-1}{2}\right)\left(\frac{4\sigma^2 a^2}{|z|^2}\right) &\approx 1,\\ p &\approx \frac{|z|^2}{2\sigma^2 a^2} + 1.\end{aligned} \tag{3.15}$$

This suggests a practical value for p would be $\bar{p} = 2\delta_2^2 + 1$. If we substitute (3.15) into (3.14) and apply the bounds δ_1 and δ_2 judiciously, we can obtain a useful estimate for the upper bound on $|R_{\bar{p}+1}|$ in (3.14):

$$|R_{\bar{p}+1}| \leq \frac{1}{2(\delta_2+1)}\left[\delta_1^{2(\delta_2+1)} + a^2\Gamma(\delta_2+2)\,\delta_2^{-(\delta_2+1)}\right], \tag{3.16}$$

which is plotted in Fig. 3.2. In practice, p is a discrete variable, but we have extended it as a continuum value and replaced the factorial with a gamma function. By inspection, the contributions from δ_2 dominate this expression.

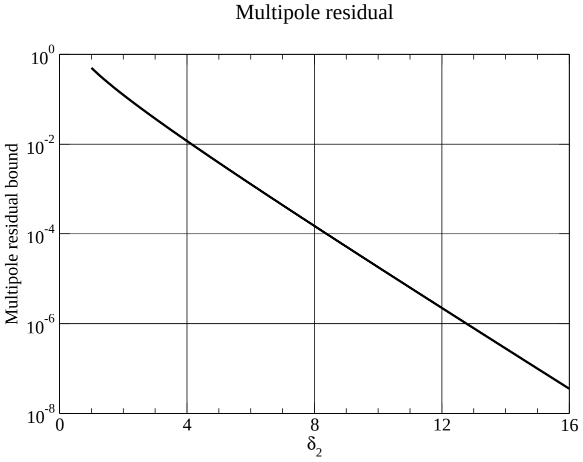

FIG. 3.2. *Exponential decay of multipole residual. This curve represents (3.16) for $a^2 = 1$ and $\delta_1 = 0$. Other values of a^2 will simply translate the curve vertically by a^2. The number of multipole terms requires varies linearly with δ_2 as in (3.15). For instance, at $\delta_2 = 8$ corresponds to 17 terms in the multipole series.*

While the convergence rate is exponential, accuracy requirements place a lower bound on the size of the finest mesh level via the parameter δ_2. When $\frac{|z|^2}{2\sigma^2 a^2}$ is too small, one must calculate the potential and its derivatives directly using the methods described in §2.

Fixing $\bar{p}$ to achieve a desire accuracy, we calculate the multipole coefficients.

$$a_n = \frac{1}{4\pi\sigma^2} \iint_{-\infty}^{\infty} \eta^n \exp\left[-\frac{\tilde{x}^2/a^2 + \tilde{y}^2 a^2}{4\sigma^2}\right] d\tilde{x} d\tilde{y}, \tag{3.17}$$

where from (3.7),

$$\varphi(z) = -\frac{i\gamma}{2\pi}\left(\ln z - \sum_{j=1}^{p} \frac{a_j}{j z^j} + R\right). \tag{3.18}$$

Integrating directly, we see that $a_1 = x_0 + iy_0 = z_0$. If $n > 1$, we develop a recurrence relation for the a_n's.

$$\begin{aligned}
a_n =& \frac{1}{4\pi\sigma^2} \iint_{-\infty}^{\infty} \eta^n \exp\left[-\frac{\tilde{x}^2/a^2 + \tilde{y}^2 a^2}{4\sigma^2}\right] d\tilde{x} d\tilde{y} \\
=& \frac{1}{4\pi\sigma^2} \iint_{-\infty}^{\infty} \left[x_0 + iy_0 + (\cos\theta + i\sin\theta)\tilde{x} + (-\sin\theta + i\cos\theta)\tilde{y}\right] \eta^{n-1} \\
& \times \exp\left[-\frac{\tilde{x}^2/a^2 + \tilde{y}^2 a^2}{4\sigma^2}\right] d\tilde{x} d\tilde{y} \\
=& z_0 a_{n-1} \\
& + \frac{1}{4\pi\sigma^2} \iint_{-\infty}^{\infty} \eta^{n-1} \left[(\cos\theta + i\sin\theta)\tilde{x} + (-\sin\theta + i\cos\theta)\tilde{y}\right] \exp\left[-\frac{\tilde{x}^2/a^2 + \tilde{y}^2 a^2}{4\sigma^2}\right] d\tilde{x} d\tilde{y}
\end{aligned}$$

Integrating by parts once, we obtain the relationship,

$$\begin{aligned} a_n =& z_0 a_{n-1} \\ &+ \frac{1}{4\pi\sigma^2}(n-1)2\sigma^2\left[a^2(\cos\theta + i\sin\theta)^2 + (-\sin\theta + i\cos\theta)^2/a^2\right] \\ &\times \iint_{-\infty}^{\infty} \eta^{n-2}\exp\left[-\frac{\tilde{x}^2/a^2 + \tilde{y}^2a^2}{4\sigma^2}\right] d\tilde{x}d\tilde{y}, \\ =& z_0 a_{n-1} + 2(n-1)\sigma^2\left[a^2(\cos\theta + i\sin\theta)^2 + (-\sin\theta + i\cos\theta)^2/a^2\right] a_{n-2}, \\ =& z_0 a_{n-1} + 2(n-1)\sigma^2 e^{2i\theta}\left(a^2 - a^{-2}\right) a_{n-2}. \end{aligned} \tag{3.19}$$

If one is differentiating to find the velocity field, one finds that $a_0 = -1$ from the logarithmic term. With (3.19), one can quickly generate the multipole coefficients for an elliptical Gaussian.

To summarize, the essential difference between classic FMM and fast summation of anisotropic Gaussians is that the latter requires that one impose a numerical length scale on the finest mesh level to achieve a specified accuracy via the partial summation of a divergent series. After finding the finest mesh level, one applies the classic FMM using (3.19) as coefficients. The computational expense of determining or manipulating the multipole coefficients for an elliptical Gaussian is no different than it would be for any other basis function.

4. Sample calculations. To verify that the method actually achieves the predicted fourth order convergence rate, we apply the full vortex method to a standard problem with a known analytic solution. The initial vorticity distribution is

$$\omega(\vec{x}, 0) = 4\exp\left(4|\vec{x}|^2\right). \tag{4.1}$$

The exact solution is

$$\omega(\vec{x}, t) = \frac{1}{\frac{1}{4} + 4t/\mathrm{Re}}\exp\left(\frac{|\vec{x}|^2}{\frac{1}{4} + 4t/\mathrm{Re}}\right), \tag{4.2}$$

where Re is the Reynolds number of the flow.

To generate the initial conditions, computational elements are laid down in a regular grid within a disk of radius 5/4 to approximate (4.1). This initial distribution can be constructed using initially isotropic or anisotropic basis functions using exact deregularization as explained in [31]. In these examples, the calculations always begin with isotropic basis functions. For all sample calculations, N/σ_0^2 remains constant where σ_0 is the initial core width and N is the number of computational elements used in the calculation. In other words, the initial overlap between computational elements is constant, $\frac{\sigma}{h} = 2$ where h is the initial grid width, as we refine the method. The initial core sizes were selected to be $\sigma_0^2 = 6.4 \times 10^{-3}$, 3.2×10^{-3}, 1.6×10^{-3} and 8.0×10^{-4}. The vortex method uses the dynamic equations provided in [29] including curvature corrections to describe the movement and deformation of the elliptical Gaussian basis functions. The velocity field and its derivatives are computed using the methods described in this paper.

The results of these calculations at different Reynolds numbers are shown in Fig. 4.1. For larger times in the simulation, the aspect ratios grow large in the domain, and the streamfunction approximation degrades to a point where the method falls back to second order spatial accuracy. The circulation weighted mean aspect ratio

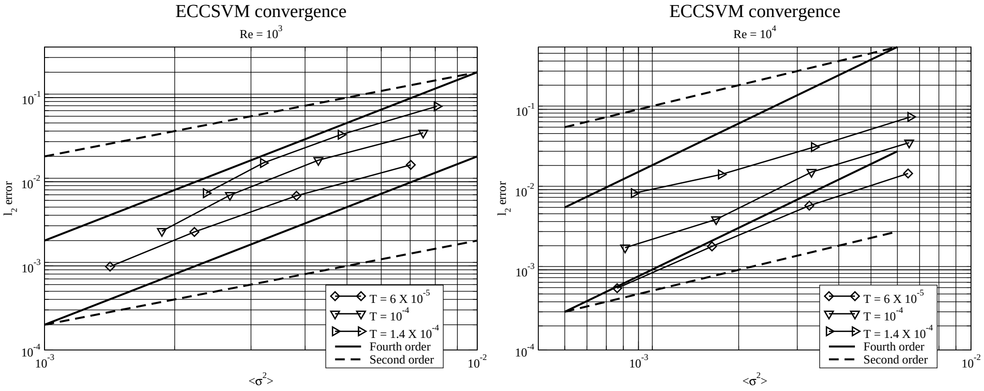

FIG. 4.1. *Convergence of ECCSVM at different Reynolds numbers with order 4 streamfunctions. As time grows, computational elements become more elongated and the Biot-Savart approximation becomes less accurate. Eventually, the velocity corrections break down, and the method reverts to second order spatial accuracy. The effective core core $\langle\sigma^2\rangle$ is a circulation-weighted average over all elements.*

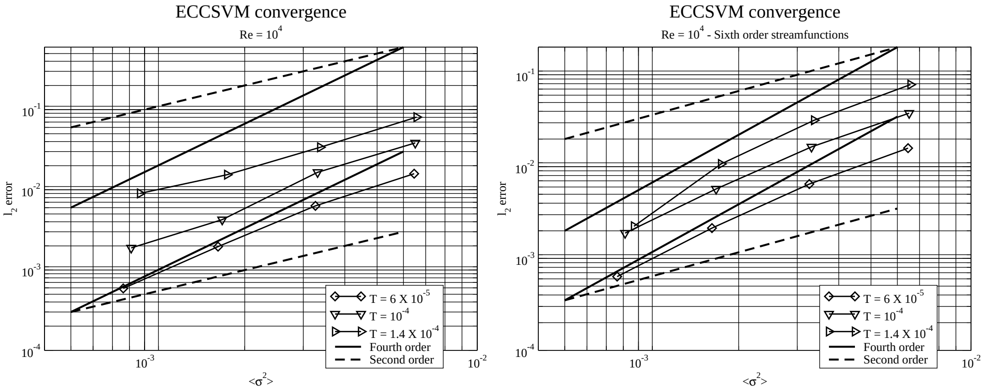

FIG. 4.2. *Convergence of ECCSVM with order 4 and 6 streamfunctions. The sixth order streamfunction permits more accurate velocity and velocity curvature computations than the forth order one as the aspect ratios grow. Thus, we see the anticipated fourth order spatial accuracy is sustained longer for the calculation using the sixth order calculation. The lower order behavior for small time is a result of interpolation error when setting the initial conditions for the computation.*

at $T = 1.4 \times 10^{-4}$ is approximately 2.8 (varying slightly from 2.7 to 2.9 depending upon σ_0^2 and the order of the streamfunction) with some aspect ratios in excess of 4.0 so that computations involve highly anisotropic elements in later phases of each calculation. The impact of using a higher accuracy streamfunction calculations is shown in Fig. 4.2. In this case, high order spatial accuracy is maintained throughout the interval shown even when basis functions are substantially elongated. Calculations were performed with and without the fast multipole method discussed in this paper using $\delta_2 = 16$, but there is no noteworthy difference save the former is substantially faster. For a single velocity evaluation on a single AMD Opteron CPU with a small problem size of $N = 1741$ ($\sigma = 6.4 \times 10^{-3}$), direct evaluation required 151 CPU seconds while the multipole computation required only 55 CPU seconds. At a problem size of $N = 13,909$ ($\sigma_0^2 = 8.0 \times 10^{-4}$), direct evaluation required 9720 seconds while the multipole computation required 496 CPU seconds.

5. Conclusions. In this paper, we have explored a direct and far-field approximation to the Biot-Savart integral for elliptical Gaussian basis functions with the purpose of developing a high spatial accuracy vortex method. The direct method uses regular perturbations and yields a consistent means of producing uniformly valid estimates of the streamfunction and its derivatives for any order of accuracy. We have shown that Greengard and Rohklin's FMM algorithm is applicable to deforming Gaussians as long as care is taken when choosing the finest mesh size, but this paper also presents a means for calculating a fine mesh size that will meet specified accuracy requirements while reducing the operation count to $O(N)$. The result is a vortex method for the 2D incompressible Navier-Stokes equations with fourth order spatial accuracy.

As discussed in the introduction, this paper has not addressed corrections, the first C in ECCSVM since it is the beyond the scope of the central problem. For large scale computations, there is a need for remeshing or splitting/merging techniques. Furthermore, splitting, merging and remeshing may serve the addition need to replace large aspect ratio elements with lower aspect ratio elements to retain accuracy in the Biot-Savart computation. Deformable elements introduce more internal degrees of freedom and so may offer some advantages inaccessible to rigid or axisymmetric elements. This remains fertile ground for future investigation.

6. Acknowledgements. The author would like to acknowledge the computational support of the National Science Foundation SCREMS DMS-0322583.

Appendix A. Algorithms for calculation of velocities and derivatives.

We represent the approximation to the kernel of the streamfunction as combination of terms:

$$\psi_1 = \frac{1}{4} \ln \left(\frac{{R_*}^2}{R^2} \right) + \sum_{p=0}^{N} \sum_{m=0}^{N} c_{m,p}^{\psi_1} \frac{{T_*}^{2m}}{{R_*}^{2(m+p)}} R^{2p}. \tag{A.1}$$

Knowing ϵ, all the c^{ψ_1}'s can be calculated.

To calculate the vector, $c^{\partial_{x_*}\psi_1}$, we represent the function as

$$\frac{\partial \psi_1}{\partial x_*} = \sum_{p=0}^{N} \sum_{m=0}^{N+1} c_{m,p}^{\partial_{x_*}\psi_1} x_* \frac{{T_*}^{2m}}{{R_*}^{2(m+p+1)}} R^{2p}, \tag{A.2}$$

and apply the following algorithm to determine $c^{\partial_{x_*}\psi_1}$ from the c^{ψ_1}.

Algorithm for $\frac{\partial}{\partial x_*}$.

```
1. Initialize the vector for $c^{\partial_{x_*}\psi_1}$ to zero.
2. $c_{0,0}^{\partial_{x_*}\psi_1} = \frac{1}{2a^2}$ from the logarithmic term in $\psi_1$.
3. $p \leftarrow 0$ to $N$
4. $m \leftarrow 0$ to $N+1$
5. If $m > 0$ then
       $c_{m,p}^{\partial_{x_*}\psi_1} = c_{m,p}^{\partial_{x_*}\psi_1} + \frac{2}{a^2} m c_{m,p}^{\psi_1}$
6.
       $c_{m,p}^{\partial_{x_*}\psi_1} = c_{m,p}^{\partial_{x_*}\psi_1} - \frac{2}{a^2} (m+p) c_{m,p}^{\psi_1}$
7. Next $m$.
8. Next $p$.
```

Both first and second derivatives increases the number of $\frac{{T_*}^{2m}}{{R_*}^{2n}}$ terms in each power of R as seen in (2.11) and (2.13), so the upper limit on the m loop is $N+1$.

Similarly, to calculate the vector, $c^{\partial_{y_*}\psi_1}$, we represent the function as

$$\frac{\partial \psi_1}{\partial y} = \sum_{p=0}^{N} \sum_{m=0}^{N+1} c_{m,p}^{\partial_{x_*}\psi_1} y_* \frac{{T_*}^{2m}}{{R_*}^{2(m+p+1)}} R^{2p}, \tag{A.3}$$

and apply the following algorithm to determine $c^{\partial_{y_*}\psi_1}$ from the c^{ψ_1}.

Algorithm for $\frac{\partial}{\partial y_*}$.

```
Initialize the vector for $c^{\partial_{y_*}\psi_1}$ to zero.
$c_{0,0}^{\partial_{y_*}\psi_1} = \frac{a^2}{2}$ from the logarithmic term in $\psi_1$.
$p \leftarrow 0$ to $N$
    $m \leftarrow 0$ to $N+1$
        If $m > 0$ then
            $c_{m,p}^{\partial_{y_*}\psi_1} = c_{m,p}^{\partial_{y_*}\psi_1} - 2a^2 m c_{m,p}^{\psi_1}$
        $c_{m,p}^{\partial_{y_*}\psi_1} = c_{m,p}^{\partial_{y_*}\psi_1} - 2a^2 (m+p) c_{m,p}^{\psi_1}$
    Next $m$.
Next $p$.
```

Approximations for second derivatives can be computed directly from ψ_1 rather than using the first derivatives as a bootstrap.

We can calculate the vector $c^{\partial_{x_*x_*}\psi_1}$ for the representation

$$\frac{\partial^2 \psi_1}{\partial {x_*}^2} = \sum_{p=0}^{N} \sum_{m=0}^{N+1} c_{m,p}^{\partial_{x_*x_*}\psi_1} \frac{{T_*}^{2m}}{{R_*}^{2(m+p+1)}} R^{2p}, \tag{A.4}$$

using the following algorithm to determine $c^{\partial_{x_*x_*}\psi_1}$ from the c^{ψ_1}.

Algorithm for $\frac{\partial^2}{\partial {x_*}^2}$.

```
Initialize the vector for $c^{\partial_{x_*x_*}\psi_1}$ to zero.
$c_{1,0}^{\partial_{x_*x_*}\psi_1} = -\frac{1}{2a^2}$ from the logarithmic term in $\psi_1$. (This step would not be included for general polynomials in $\frac{{T_*}^{2m}}{{R_*}^{2n}}$.)
$p \leftarrow 0$ to $N$
    $m \leftarrow 0$ to $N+1$
        If $m > 0$
            $c_{m-1,p}^{\partial_{x_*x_*}\psi_1} = c_{m-1,p}^{\partial_{x_*x_*}\psi_1} + \frac{2}{a^2}(m+p)(2p-m) c_{m,p}^{\psi_1}$
        If $m > 1$
            $c_{m-2,p}^{\partial_{x_*x_*}\psi_1} = c_{m-2,p}^{\partial_{x_*x_*}\psi_1} + \frac{2}{a^2} m(m-1) c_{m,p}^{\psi_1}$
        $c_{m,p}^{\partial_{x_*x_*}\psi_1} = c_{m,p}^{\partial_{x_*x_*}\psi_1} + \frac{2}{a^2}(p+m)(p-m) c_{m,p}^{\psi_1}$
        If $m < N-1$
            $c_{m+1,p}^{\partial_{x_*x_*}\psi_1} = c_{m+1,p}^{\partial_{x_*x_*}\psi_1} + \frac{2}{a^2}(p+m) c_{m,p}^{\psi_1}$
    Next $m$.
Next $p$.
```

We can calculate the vector $c^{\partial_{y_* y_*}\psi_1}$ for the representation

$$\frac{\partial^2 \psi_1}{\partial {y_*}^2} = \sum_{p=0}^{N} \sum_{m=0}^{N+1} c_{m,p}^{\partial_{y_* y_*}\psi_1} \frac{{T_*}^{2m}}{{R_*}^{2(m+p+1)}} R^{2p}, \tag{A.5}$$

using the following algorithm to determine $c^{\partial_{y_* y_*}\psi_1}$ from the c^{ψ_1}.

```
Algorithm for $\frac{\partial^2}{\partial {y_*}^2}$.
 1. Initialize the vector for $c^{\partial_{y_* y_*}\psi_1}$ to zero.
 2. $c_{1,0}^{\partial_{y_* y_*}\psi_1} = \frac{1}{2}a^2$ from the logarithmic term in $\psi_1$. (This step
    would not be included for general polynomials in $\frac{{T_*}^{2m}}{{R_*}^{2n}}$.)
 3. $p \leftarrow 0$ to $N$
 4. $m \leftarrow 0$ to $N+1$
 5. If $m > 0$
        $c_{m-1,p}^{\partial_{y_* y_*}\psi_1} = c_{m-1,p}^{\partial_{y_* y_*}\psi_1} + 2a^2(m+p)(2p-m)c_{m,p}^{\psi_1}$
 6. If $m > 1$
        $c_{m-2,p}^{\partial_{y_* y_*}\psi_1} = c_{m-2,p}^{\partial_{y_* y_*}\psi_1} + 2a^2 m(m-1)c_{m,p}^{\psi_1}$
 7.
        $c_{m,p}^{\partial_{y_* y_*}\psi_1} = c_{m,p}^{\partial_{y_* y_*}\psi_1} + 2a^2(p+m)(p-m)c_{m,p}^{\psi_1}$
 8. If $m < N-1$
        $c_{m+1,p}^{\partial_{y_* y_*}\psi_1} = c_{m+1,p}^{\partial_{y_* y_*}\psi_1} - 2a^2(p+m)c_{m,p}^{\psi_1}$
 9. Next $m$.
10. Next $p$.
```

We can calculate the vector $c^{\partial_{x_* y_*}\psi_1}$ for the representation

$$\frac{\partial^2 \psi_1}{\partial x_* \partial y_*} = \sum_{p=0}^{N} \sum_{m=0}^{N+1} c_{m,p}^{\partial_{x_* y_*}\psi_1} x_* y_* \frac{{T_*}^{2m}}{{R_*}^{2(m+p+2)}} R^{2p}, \tag{A.6}$$

using the following algorithm to determine $c^{\partial_{y_* y_*}\psi_1}$ from the c^{ψ_1}.

```
Algorithm for $\frac{\partial^2}{\partial x_* \partial y_*}$.

    1. Initialize the vector for $c^{\partial_{y_* y_*}\psi_1}$ to zero.
    2. $c_{0,0}^{\partial_{x_* y_*}\psi_1} = -1$ from the logarithmic term in $\psi_1$. (This step
       would not be included for general polynomials in $\frac{{T_*}^{2m}}{{R_*}^{2n}}$.)
    3. $p \leftarrow 0$ to $N$
    4. $m \leftarrow 0$ to $N+1$
    5. If $m > 1$
           $c_{m-2,p}^{\partial_{x_* y_*}\psi_1} = c_{m-2,p}^{\partial_{x_* y_*}\psi_1} - 4m(m-1)c_{m,p}^{\psi_1}$
    6.
           $c_{m,p}^{\partial_{x_* y_*}\psi_1} = c_{m,p}^{\partial_{x_* y_*}\psi_1} + 4(m+p)(m+p+1)c_{m,p}^{\psi_1}$
    7. Next $m$.
    8. Next $p$.
```

Higher order derivatives are calculated using first and second derivatives as building blocks and using these building blocks in the proper order. For instance, to calculate $\frac{\partial^3\psi_1}{\partial {x_*}^3}$, we would begin with ψ_1 and calculate $\frac{\partial^2\psi_1}{\partial {x_*}^2}$ which has the form given in (A.4). Then, we can apply the same $\frac{\partial}{\partial x_*}$ operation given before but without step 2 because there is no logarithmic term and the index on the $\frac{{T_*}^{2m}}{{R_*}^{2n}}$ terms is shifted by one. Therefore, we develop the following representation,

$$\frac{\partial\psi_1}{\partial {x_*}^3} = \sum_{p=0}^{N}\sum_{m=0}^{N+1} c_{m,p}^{\partial_{x_* x_* x_*}\psi_1} x_* \frac{{T_*}^{2m}}{{R_*}^{2(m+p+2)}} R^{2p}, \tag{A.7}$$

using the algorithm below.

```
Algorithm for $\frac{\partial}{\partial x_*}$.

    1. Initialize the vector for $c^{\partial_{x_* x_* x_*}\psi_1}$ to zero.
    2. $p \leftarrow 0$ to $N$
    3. $m \leftarrow 0$ to $N+1$
    4. If $m > 0$ then
           $c_{m,p}^{\partial_{x_* x_* x_*}\psi_1} = c_{m,p}^{\partial_{x_* x_* x_*}\psi_1} + \frac{2}{a^2} m c_{m,p}^{\partial_{x_* x_*}\psi_1}$
    5.
           $c_{m,p}^{\partial_{x_* x_* x_*}\psi_1} = c_{m,p}^{\partial_{x_* x_* x_*}\psi_1} - \frac{2}{a^2}(m+p+1) c_{m,p}^{\partial_{x_* x_*}\psi_1}$
    6. Next $m$.
    7. Next $p$.
```

REFERENCES

[1] C. Anderson, *A method of local corrections for computing the velocity field due to a distribution of vortex blobs*, J. Comp. Phys., 62 (1986), pp. 111–123.

[2] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 923–947.

[3] L. Barba, *Discussion: "three-dimensional vortex method for gas-particle two-phase compound round jet" Uchiyama, T. Fukase, A., 2005, ASME J. Fluids Engg., 127, pp. 32-40.* Submitted.

[4] L. A. Barba, A. Leonard, and C. B. Allen, *Advances in viscous vortex methods - meshless spatial adaption based on radial basis functions*, Int. J. Num. Methods in Fluids, 47 (2005), pp. 387–421.

[5] H. Cheng, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm in three dimensions*, J. Comp. Phys., 155 (1999), pp. 468–498.

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

[7] C. I. Draghicescu, *An efficient implementation of particle methods for the incompressible Euler equations.*, SIAM J. Num. Anal., 31 (1994), pp. 1090–1108.

[8] ———, *A fast algorithm for vortex blob interactions.*, Journal of Computational Physics, 116 (1995), pp. 69–78.

[9] C. I. Draghicescu and M. Draghicescu, *A fast algorithm for vortex blob interactions*, J. Comput. Phys., 116 (1995), pp. 69–78.

[10] A. Dutt, M. Gu, and V. Rokhlin, *Fast algorithms for polynomial interpolation, intergration and differentiation*, SIAM J. Numer. Anal., 33 (1996), pp. 1689–1711.

[11] Z. Gimbutas and V. Rokhlin, *A generalized fast multipole method for nonoscillatory kernels*, SIAM J. Sci. Comput., 24 (2002), pp. 796–817.

[12] C. Greengard, *The core spreading vortex method approximates the wrong equation*, J. Comp. Phys., 61 (1985), pp. 345–348.

[13] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.

[14] R. Hrycak and V. Rokhlin, *An improved fast multipole algorithm for potential fields*, SIAM J. Sci. Comput., 19 (1998), pp. 1804–1826.

[15] N. J. Huang, *Diffusion via splitting and remeshing via merging in vortex methods*, Int. J. Num. Methods Fluids, 48 (2005), pp. 521 – 539.

[16] 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.

[17] O. M. Knio, L. Collorec, and D. Juvé, *Numerical study of sound emission by 2d regular and chaotic vortex configurations*, J. Comp. Phys., 116 (1995), pp. 226–246.

[18] H. Lamb, *Hydrodynamics*, Cambridge University Press, 1993, ch. IV, VII, pp. 84–86, 232–233.

[19] A. Leonard, *Vortex methods for flow simulation*, J. Comp. Phys., 37 (1980), pp. 289–335.

[20] ———, *AIAA 97-0204: Large-eddy simulation of chaotic convection and beyond*, in 35th Aerospace Sciences Meeting & Exhibit, American Institute of Aeronautics and Astronautics, 1997.

[21] 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.

[22] P. Moeleker and A. Leonard, *Lagrangian methods for the tensor-diffusivity subgrid model*, J. Comp. Phys., 167 (2001), pp. 1–21.

[23] 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.

[24] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren, *Vortex methods for direct numerical simulation of bluff body flows: Application to the sphere at Re = 300, 500, 1000.*, J. Comp. Phys., 178 (2002), pp. 427–463.

[25] G. Riccardi and R. Piva, *Motion of an elliptical vortex under rotating strain: conditions for asymmetric merging*, Fluid Dyn. Res., 23 (1998), pp. 63–88.

[26] L. Rosenhead, *The point vortex approximation of a vortex sheet*, Proceedings of the Royal Society of London Series A, 134 (1931), pp. 170–192.

[27] 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.

[28] ———, *Merging computational elements in Lagrangian simulations*, SIAM J. Sci. Comp., 18 (1997), pp. 1014–1027.

[29] ———, *Achieving high-order convergence rates with deforming basis functions.*, SIAM J. Sci. Comput., 26 (2005), pp. 885–906.

[30] ———, *A comparative study of lagrangian methods using axisymmetric and deforming blobs. deforming blobs*, SIAM J. Sci. Comput., (2005). To appear. Tech report 2003-9 available online at http://www.math.udel.edu/research/techrept/tech_2003.html.

[31] L. F. Rossi, J. F. Lingevitch, and A. J. Bernoff, *Quasi-steady monopole and tripole at-*

tractors for relaxing vortices, Physics of Fluids, 9 (1997), pp. 2329–2339.

[32] 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.

[33] B. N. Shashikanth and P. K. Newton, *Geometric phases for corotating elliptical vortex patches*, J. Math. Phys., 41 (2000), pp. 8148–8162.

[34] D. Shiels, *Simulation of controlled bluff body flow with a viscous vortex method*, PhD thesis, California Institute of Technology, Pasadena, CA, 1998.

[35] D. Shiels and A. Leonard, *Investigation of a drag reduction on a circular cylinder in rotary oscillation*, J. Fluid Mech., 431 (2001), pp. 297–322.

[36] D. Sipp, D. Fabre, S. Michelin, and L. Jacquin, *Stability of a vortex with a heavy core*, J. Fluid Mech., 526 (2005), pp. 67–76.

[37] Z.-H. Teng, *Elliptic-vortex method for incompressible flow at high Reynolds number.*, J. Comp. Phys., 46 (1982), pp. 54–68.

[38] ———, *Variable-elliptical-vortex method for incompressible flow simulation.*, J. Comp. Math., 4 (1986), pp. 255–262.

[39] ———, *Convergence of the variable-elliptic-vortex method for Euler equations*, SIAM J. Num. Anal., 32 (1995), pp. 754–774.

[40] T. Uchiyama and M. Naruse, *Numerical simulation of gas-particle two-phase mixing layer by vortex method*, Powder Technology, 125 (2002), pp. 111–121.

[41] ———, *Vortex simulation of slit nozzle gas-particle two-phase jet*, Powder Technology, 131 (2003), pp. 156–165.

[42] T. Uchiyama and T. Okita, *Numerical prediction of a plume diffusion field around a circular cylinder by the particle method*, Advances in Environmental Research, 7 (2003), pp. 573–581.

[43] L. van Dommelen and E. A. Rundensteiner, *Fast, adaptive summation of point forces in the two-dimensional poisson equation*, Journal of Computational Physics, 83 (1989), pp. 126–147.

[44] I. Yasuda and G. R. Flierl, *Two-dimensional asymmetric vortex merger: merger dynamics and critical merger distance*, Dyn. Atmos. Oceans, 26 (1997), pp. 159–181.