INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:387–421
Published online 30 November 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.811

Advances in viscous vortex methods—meshless spatial adaption based on radial basis function interpolation

L. A. Barba[1,*,†], A. Leonard[2] and C. B. Allen[1]

[1]*Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, U.K.*
[2]*Graduate Aeronautical Laboratories, California Institute of Technology, Pasadena CA 91125, U.S.A.*

SUMMARY

Vortex methods have a history as old as finite differences. They have since faced difficulties stemming from the numerical complexity of the Biot–Savart law, the inconvenience of adding viscous effects in a Lagrangian formulation, and the loss of accuracy due to Lagrangian distortion of the computational elements. The first two issues have been successfully addressed, respectively, by the application of the fast multipole method, and by a variety of viscous schemes which will be briefly reviewed in this article. The standard method to deal with the third problem is the use of remeshing schemes consisting of tensor product interpolation with high-order kernels. In this work, a numerical study of the errors due to remeshing has been performed, as well as of the errors implied in the discretization itself using vortex blobs. In addition, an alternative method of controlling Lagrangian distortion is proposed, based on ideas of radial basis function (RBF) interpolation (briefly reviewed here). This alternative is formulated grid-free, and is shown to be more accurate than standard remeshing. In addition to high-accuracy, RBF interpolation allows core size control, either for correcting the core spreading viscous scheme or for providing a variable resolution in the physical domain. This formulation will allow in theory the application of error estimates to produce a truly adaptive spatial refinement technique. Proof-of-concept is provided by calculations of the relaxation of a perturbed monopole to a tripole attractor. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: vortex method; radial basis functions; core spreading; viscous flow simulation; meshless methods

1. INTRODUCTION

Vortex methods for the simulation of incompressible flows correspond to a numerical approach with three fundamental features. First, the Navier–Stokes or Euler equations are formulated in terms of vorticity and so the spatial discretization is carried out over the vorticity field instead of the velocity field. Second, making use of one of Helmholtz' theorems which

*Correspondence to: L. A. Barba, Department of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, U.K.
†E-mail: L.A.Barba@bris.ac.uk

Received 10 October 2003
Revised 20 September 2004
Accepted 14 October 2004

states the correspondence of vorticity elements with material fluid elements, the computational elements are Lagrangian and so convect with the fluid velocity. And third, to obtain the fluid velocity one makes use of the fact that the vorticity, defined as $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, can be inverted giving the velocity $\mathbf{u}$ as an integral over the vorticity field. This is the Biot–Savart law in vorticity kinematics, which allows to completely describe the flow field by tracking vorticity elements.

The features described above generate both the most desirable aspects of vortex methods, as well as their most serious difficulties. Describing the flow in terms of vorticity is desirable due to the intuitive power of visualizing the vorticity field, especially in complex and unsteady flows. Another advantage is the fact that the pressure drops out of the governing equation, and thus only needs to be solved for when and where force measurements are desired. In addition, as the vortex method literature profusely extols, the fact that the vorticity field is predominantly compact means that smaller computational domains can be used, in comparison with primitive variable formulations, and boundary conditions at infinity can be automatically satisfied. In contrast, satisfying the free-space boundary condition of external flows can be a delicate matter in grid-based methods with truncated flow domains. Furthermore, the Lagrangian vortex particles convect without numerical dissipation, as the non-linear term of the Euler or Navier–Stokes equations is traded for a set of ordinary differential equations for the particle trajectories. This is, again, in contrast with grid-based schemes which invariably suffer from numerical dissipation. Finally, the essentially grid-free nature of the vortex particle method is itself an advantage, as grid-generation is often one of the most expensive processes in CFD (see, for example, Reference [1]).

Vortex methods have been around almost as long as finite differences and the earliest methods of computational mathematics. Indeed, the seminal work of Praeger [2] with vortex distributions on surfaces is the origin of panel methods—widely used in the aeronautics industry to this day—whereas Rosenhead's work on the calculation of vortex sheets with the point vortex method [3] was of such great consequence that it is still very much cited today. But vortex methods have, in spite of this, suffered the reputation of being mostly coarse attempts at modelling complex vortical flows (admittedly difficult to tackle with traditional CFD methods as well). The main difficulties for vortex methods to be accepted in the mainstream of computational fluid dynamics have been threefold: (i) the numerical complexity of calculating the velocity using the Biot–Savart law (requiring $O(N^2)$ operations for N vortex elements); (ii) the inconvenience of adding viscous effects in a Lagrangian formulation; and, (iii) the effect of the Lagrangian discretization itself producing a loss of accuracy due to its distortion. The first of these difficulties has been successfully addressed by the application of the fast multipole method [4] for the calculation of the particle velocities, whereas some workers have bypassed the problem with mixed Eulerian–Lagrangian formulations such as the vortex-in-cell method [5], although at the cost of adding mesh-dependence and interpolation errors. The addition of viscous effects, on the other hand, has benefited from profuse research, there being at least seven proven schemes (these will be reviewed briefly below, in Section 3) with varied degree of success. The loss of accuracy due to Lagrangian distortion of the particles, finally, has been dealt with by the application of remeshing schemes, which utilize high-order interpolation kernels on a Cartesian tensor product formulation. Although the standard remeshing schemes have made long-time, accurate calculations of complex flows possible, they have caused quite a bit of controversy as they add a mesh to an otherwise meshless method. In addition, they do introduce some interpolation error, generally

accepted as tolerable. As one wishes to simulate flows at higher Reynolds numbers, however, increased resolution becomes crucial and the interpolation error may be a limitation. Or, viewed from another perspective, a more accurate method may allow for reduced problem sizes (i.e. smaller number of vortex particles for a given accuracy) at high Reynolds numbers.

In this article, after first presenting the formulation of vortex (blob) methods in Section 2 and the viscous schemes in Section 3, a numerical study of the accuracy of the vortex method will be presented. This study first concentrates on the spatial discretization, that is, on answering the question of how accurately one can discretize the vorticity field with vortex blobs. As detailed in Section 4, the main parameter which governs this accuracy is the overlap ratio, defined as the ratio between inter-particle spacing h and particle size σ. This is by no means new, as convergence analyses of the vortex method—initiated more than 20 years ago—have been based on the assumption that the overlap ratio is kept smaller than 1, and hence workers recognize that overlap of vortex blobs must be maintained throughout a calculation. But presently it is shown, in practice, how several orders of magnitude in the accuracy are lost as the overlap ratio increases passing through the value of 1. More importantly, one is reassured by these results that high-resolution is indeed possible with the vortex blob method, as accuracy of the order of Intel double precision is attainable when the overlap ratio, h/σ, is about 0.7 or smaller, for a Gaussian blob function.

Subsequently, this work studies in Section 5 the accuracy implications of the Lagrangian distortion of the particle field. It is a consequence of the strong dependence of discretization accuracy on the overlap ratio that the Lagrangian advection of vortex blobs will lead to an increase of errors. Here, this is illustrated using two test problems which have an analytical solution, so that errors are precisely quantified. Next, the improvement that can be obtained using the standard remeshing schemes with high-order interpolation kernels is demonstrated. It is seen, however, how with a very accurate initialization the error introduced upon the first event of location processing becomes apparent. In other words, although the standard techniques do allow for long-time calculations with tolerably small error, they also impose a limitation on the accuracy that one can obtain with the vortex method, evident in what we dub an 'initial remesh error'.

Remeshing—or what some workers prefer to call 'redistribution', unfortunately introducing confusion with the vortex redistribution method, a diffusion scheme—is by no means the only method to control the errors due to Lagrangian distortion of the particle field. One of the early works analysing the accuracy of vortex methods [6] simply proposed to choose larger initial particle core sizes (i.e. smaller values of the overlap ratio) for longer time calculations, so that by merely a denser initialization it would be possible to maintain overlap. A more useful proposal was the 'rezoning' techniques [7], which will be seen below to be applicable in particular with higher-order blob or cutoff functions. In addition, a technique which attempts to correct the Lagrangian errors not by re-locating the vortex particles to a uniform mesh, but by changing their circulation strengths to better approximate the continuous vorticity field, is provided by Beale's method of circulation processing [8]. These schemes for spatial adaption in vortex methods will be discussed in Section 6.

The focal contribution of this work is the formulation of a spatial adaption scheme which does not require an ordered grid, and which is more accurate than other methods, including standard remeshing schemes with high-order kernels. The proposed method is based on ideas from radial basis function (RBF) interpolation, a technique for solving multivariate scattered

data interpolation problems. In fact, the similarity between the vortex blob discretization and Beale's method of circulation processing, and scattered data interpolation has been alluded to before [8, 9]. After a brief presentation of RBF interpolation fundamentals, Section 7 presents the application of this technique for spatial adaption in vortex methods. Finally, Section 8 presents results of numerical computations using this method on a perturbed monopole which relaxes to a tripole attractor, which contribute to validate the proposed scheme by comparison with similar computations in Reference [10].

2. THE VORTEX BLOB METHOD

Let $\mathbf{u}(\mathbf{x},t)$ be the velocity field and $\boldsymbol{\omega}(\mathbf{x},t)=\nabla\times\mathbf{u}(\mathbf{x},t)$ the vorticity field. Taking the curl of the momentum equation and considering an incompressible fluid, the vorticity transport equation is obtained. This is the governing equation in vortex methods, which for three-dimensional flows is

$$\frac{\partial\boldsymbol{\omega}}{\partial t}+\mathbf{u}\cdot\nabla\boldsymbol{\omega}=\boldsymbol{\omega}\cdot\nabla\mathbf{u}+\nu\Delta\boldsymbol{\omega} \tag{1}$$

The assumptions in (1) are: constant density flow, conservative body forces, inertial frame of reference, and unbounded domain. In the case of a two-dimensional and inviscid flow the right-hand side of (1) is zero and the governing equation reduces to the simple form $\mathrm{D}\omega/\mathrm{D}t=0$, where $\mathrm{D}/\mathrm{D}t$ stands for the material derivative. This corresponds to the basic formulation of vortex methods, for which clearly a Lagrangian method based on elements of vorticity is natural and ideal. Based on this simplest of formulations, the vortex method historically found its first successful applications in the simulation of phenomena governed by the 2D Euler equations. Subsequently, vortex methods have been extended to three-dimensional flow by including the vortex stretching/tilting term, and have incorporated the presence of internal boundaries by using vortex sheet formulations in the inviscid case and vorticity generation models with boundary elements for the viscous case. Viscous effects were added first by the random walk method [11], but a number of the so-called 'deterministic' viscous schemes have been proposed and tested during the last two decades.

In the vortex blob method, the elements are identified by: a position vector, $\mathbf{x}_i$; a strength vector (vorticity $\times$ volume) of circulation; and a core size, σ_i. The discretized vorticity field is expressed as the sum of the vorticities of the vortex elements in the following way:

$$\boldsymbol{\omega}(\mathbf{x},t)\approx\boldsymbol{\omega}^h(\mathbf{x},t)=\sum_{i=1}^{N}\boldsymbol{\Gamma}_i(t)\zeta_{\sigma_i}(\mathbf{x}-\mathbf{x}_i(t)) \tag{2}$$

where $\boldsymbol{\Gamma}_i$ corresponds to the vector (scalar in 2D) circulation strength of particle i. In the blob version of the vortex method—in contrast to point vortices—the elements have a non-zero, often uniform, core size $\sigma_i=\sigma$ and a characteristic distribution of vorticity ζ_{σ_i}, commonly called the cutoff function. Frequently, the blob cutoff function is a Gaussian distribution, such as (in 2D):

$$\zeta_\sigma(\mathbf{x})=\frac{1}{2\pi\sigma^2}\,\mathrm{e}^{-|\mathbf{x}|^2/2\sigma^2} \tag{3}$$

The vortex elements are assumed to convect without deformation with the local velocity, which is approximated by the Biot–Savart law:

$$\mathbf{u}(\mathbf{x},t)=\int(\nabla\times\mathbf{G})(\mathbf{x}-\mathbf{x}')\omega(\mathbf{x}',t)\,\mathrm{d}\mathbf{x}'=\int\mathbf{K}(\mathbf{x}-\mathbf{x}')\omega(\mathbf{x}',t)\,\mathrm{d}\mathbf{x}'=(\mathbf{K}*\omega)(\mathbf{x},t) \tag{4}$$

where $\mathbf{K}=\nabla\times\mathbf{G}$ is the Biot–Savart kernel, $\mathbf{G}$ is the Green's function for the Poisson equation, and $*$ represents convolution. For example, in 2D the Biot–Savart law is written explicitly as

$$\mathbf{u}(\mathbf{x},t)=\frac{-1}{2\pi}\int\frac{(\mathbf{x}-\mathbf{x}')\times\omega(\mathbf{x}',t)\hat{\mathbf{k}}}{|\mathbf{x}-\mathbf{x}'|^2}\,\mathrm{d}\mathbf{x}' \tag{5}$$

Finally, the Lagrangian formulation of the (viscous) vortex method in two dimensions can be expressed in the following system of equations:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}=\mathbf{u}(\mathbf{x}_i,t)=(\mathbf{K}*\omega)(\mathbf{x}_i,t) \tag{6}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t}=\nu\nabla^2\omega+\text{B.C.} \tag{7}$$

The complete numerical method is defined by Equations (6) and (7) which express that the implementation consists in integrating the particle trajectories due to the local fluid velocity, while the velocity is obtained from the vorticity using the Biot–Savart law. The vorticity field evolves due to the effects of viscosity, both in the free-stream and on the boundaries (no-slip condition, denoted by 'B.C.'). The viscous effects in the free-stream are enforced by one of a variety of viscosity schemes available for vortex methods (see next section), while the effects due to solid boundaries are traditionally accounted for by generation of vorticity implemented in a version of the boundary element method.

Comprehensive reviews of the development of vortex methods and their applications can be found in References [12–16]. The favoured applications of vortex methods may be found in the simulation of unsteady flows that exhibit concentrated areas of strong vorticity. For example, bluff body flows and wake-induced phenomena are clear candidates for being tackled with vortex methods. Also, separated or oscillating flows around airfoils, and problems like dynamic stall and vortex-blade interaction—especially as higher Reynolds number computations become feasible. In these types of application problems it is more likely that the special features of vortex methods, e.g. the Lagrangian and grid-free nature of the formulation and the concentration of computational elements only on the compact support of the vorticity field, will result in advantages over using other more traditional methods.

3. VISCOUS SCHEMES FOR VORTEX METHODS

The Lagrangian, grid-free approach in vortex methods has caused quite a bit of difficulty in making the extension to viscous flows efficiently and accurately. The first successful addition of viscosity was made with a random walk [11] of the vortex positions (now called random vortex method, RVM). In this approach, diffusion is simulated with a Brownian motion of the particles, based on the probabilistic interpretation of the diffusion equation.

The random vortex method is formulated as a fractional step method, i.e. the viscous and inviscid parts of the governing equations are taken into account as successive sub-steps. More sub-steps are involved in higher-order schemes, but the basic two-step viscous splitting algorithm is second-order accurate at each time step and first-order overall (irrespective of the time-stepping scheme used). So, in addition to other sources of error, the RVM suffers from operator splitting error. The RVM error estimate [17] is of order $\sqrt{\nu/N}$, which in 2D corresponds to first order in the particle spacing, h, for a regular grid, and to $O(h^{3/2})$ in a 3D rectangular discretization. Convergence has been proved [18, 19], but its slow rate means that a large number of particles are required to obtain reasonable accuracy. Nevertheless, it has the advantage of preserving the Lagrangian and grid-free formulation of vortex methods, it is local and easily parallelizable, and it continues to find extensive use to this day, especially in engineering applications. In spite of this, the disadvantages are important: only slightly viscous flows can be modelled (lower Reynolds number limit is about 100); pressures and forces can only be obtained by averaging or smoothing over several time steps; there is an error in the mean size and position of the vortex system; and it has low convergence rate and low accuracy, as mentioned.

Contemporary to the introduction of the RVM, it was recognized that vortex blobs could be allowed to grow in time to simulate viscosity [20, 12], which came to be known as the core spreading method. The idea of core spreading can be explained for the 2D case using as analogy the classical exact solution of the Navier–Stokes equations termed 'spreading line vortex' [21, p. 204]. In this problem the vorticity is given by

$$\omega(\mathbf{x},t)=\frac{\Gamma}{4\pi\nu t}\,\mathrm{e}^{-|\mathbf{x}|^2/4\nu t} \tag{8}$$

Consider again the approximate vorticity given by the discretized form (2), but in two dimensions (where the strengths are scalar), and write it slightly differently to express the fact that the core function now is dependent on time,

$$\omega(\mathbf{x},t)\approx\omega^h(\mathbf{x},t)=\sum_{i=1}^{N}\Gamma_i\zeta(\mathbf{x}-\mathbf{x}_i(t),t) \tag{9}$$

The core function is now chosen to be the solution of the heat equation with initial data ζ_o,

$$\zeta(\mathbf{x},t)=\frac{1}{4\pi\nu t}\int \mathrm{e}^{-(\mathbf{x}-\mathbf{y})^2/4\nu t}\zeta_o(\mathbf{y})\,\mathrm{d}\mathbf{y}=(G*\zeta_o)(\mathbf{x}) \tag{10}$$

where G is the heat kernel. If the initial distribution function ζ_o is a Dirac delta, then:

$$\zeta(\mathbf{x},t)=\frac{1}{4\pi\nu t}\mathrm{e}^{-|\mathbf{x}|^2/4\nu t} \tag{11}$$

Comparing with (8), the discretized vorticity field in 2D can be seen as a superposition of spreading line vortices of different circulation strengths Γ_i. The core spreading vortex method is then formulated to satisfy identically the viscous part of the vorticity equation by growing σ^2 linearly according to

$$\frac{\mathrm{d}\sigma^2}{\mathrm{d}t}=2\nu \tag{12}$$

Thus, the method is expressed in the simple algorithmic rule,

$$\sigma_i^2(t+\Delta t) = \sigma_i^2(t) + 2\nu\Delta t, \quad i = 1 \cdots N \tag{13}$$

This method is utterly simple to implement, it is fully localized and grid-free, and easily parallelizable—as is the RVM, but core spreading is deterministic so in principle allows for better error control and faster convergence than the RVM. However, research and use of the core spreading method stalled when the method was declared inconsistent as Greengard [22] gave mathematical proof that it converges to an equation that differs from the Navier–Stokes equation in the convection term. The consistency error of core spreading is caused by the advection without deformation of larger and larger vortex blobs as they spread. About a decade later [23], a splitting scheme was finally proposed to alleviate this problem and proved to be convergent. Research on the core spreading vortex method was effectively 'resurrected', but blob splitting does introduce some numerical diffusion; its errors are in fact comparable to the random vortex method.

Whilst the core spreading method remained in discredit, a number of other deterministic viscous schemes were developed. Possibly the most prevalent of these is particle strength exchange (PSE), based on general particle methods [24]. In this method, the Laplacian at a particle's location is approximated based on nearby particles by an integral operator (provided that the smoothing function η satisfies certain moment conditions, see Reference [25, p. 145]), as follows:

$$\nabla^2\omega(\mathbf{x}) \approx \frac{2}{\sigma^2}\int \eta_\sigma(\mathbf{x}-\mathbf{x}')[\omega(\mathbf{x}')-\omega(\mathbf{x})]\,\mathrm{d}\mathbf{x}' = \int G_\sigma(|\mathbf{x}-\mathbf{x}'|)[\omega(\mathbf{x}')-\omega(\mathbf{x})]\,\mathrm{d}\mathbf{x}' \tag{14}$$

Using a Gaussian smoothing function, one has

$$G_\sigma(\mathbf{x}) = \frac{2}{\pi\sigma^2}\,\mathrm{e}^{-|\mathbf{x}|^2/2\sigma^2} \tag{15}$$

so that when $\sigma^2 = 2\nu\delta t$, (15) is equivalent to the heat kernel. In this case, the accuracy of the approximation given by (14) is $O(\sigma^2)$. The integral operator in (14) is discretized by quadrature, using as quadrature points the locations of the particles. In principle, the formulation of PSE is grid-free, but the fact that the accuracy relies strongly on the quadrature rules used for the discretized integral means that in practice the method hinges on having nearly uniformly spaced particle locations. For this reason, the extensive implementation of the PSE method has promoted the development and widespread use of remeshing schemes in vortex methods [26–30]. This has caused a bit of debate, as some workers maintain that the grid-free nature of the vortex method is undermined when remeshing schemes relying on regular particle grids are applied, sometimes as often as every time step. Indeed, if this is the case, as pointed out by G Winckelmans (private communication, 2002), then there may not be much difference between PSE with remeshing and vortex-in-cell methods (which nowadays utilize the same interpolation kernels as particle remeshing). In both cases, each interpolation step introduces some numerical diffusion, although these errors are generally considered acceptable and indeed a number of remarkable results have been obtained on unsteady wake flows, e.g. References [31, 32] using PSE with remeshing, and References [33, 34] using the vortex-in-cell approach. In spite of this, it is possible that the interpolation errors in both of these approaches impose a limitation on the accuracy of the vortex method,

especially at high Reynolds numbers where the small physical viscosity might be overwhelmed by the numerical diffusion effects.

The heavy reliance of the PSE method on remeshing schemes motivated the development of an alternative which is known as the vortex redistribution method, VRM [35]. This method is based on particle circulation exchange as well, but the formulation is different from PSE as it does not use integral operators. The algorithm hinges on finding fractions of each particle's circulation that will be 'redistributed' among its neighbours, these being defined as those particles within a maximum distance in the order of the typical diffusion distance, $h_v = \sqrt{\nu \Delta t}$. The 'redistribution fractions' f_{ij}^n are solved for by assembling a system of equations which is formulated so that locally there is conservation of circulation, linear and angular momentum, and enforcing positivity of the solution: $f_{ij}^n \geqslant 0$. Whenever it is encountered that the linear systems for the f_{ij}^n have no acceptable solution, an *ad hoc* algorithm is used that inserts new vortex particles within the neighbourhood in question. The number of vortices thus can increase without apparent bound when the Reynolds number is increased, and vortex particle merging is sometimes used to alleviate this problem.

The authors of the VRM claim that the advantage of the method is that it retains the grid-free nature of vortex methods, because it is not needed that the particles be in an ordered distribution (in contrast to PSE). This is achieved, however, at the cost of solving N underdetermined systems for N particles, at each time step. The size of the systems is determined by the redistribution influence neighbourhood, $|\mathbf{x}_j - \mathbf{x}_i| \leqslant R\sqrt{\nu \Delta t}$, involving an empirically chosen parameter R. Then again, as will be discussed in the following two sections, no matter the viscous scheme, the accuracy of the vortex method in general does depend on preserving overlap throughout a calculation. The vortex insertion algorithm in the VRM does provide some form of spatial adaption, but with no overlap control. In addition, there may be some numerical diffusion involved in the vortex merging processes, and moreover, viscous splitting error is present.

There are yet other deterministic viscous schemes for vortex methods, namely: Fishelov's method, diffusion velocity method, and least-squares and triangulated vortex methods. Fishelov's [36] suggestion was to approximate the second-order derivatives in the Laplacian by explicit differentiation of the cutoff function. This idea had already been applied in Reference [37] to obtain the stretching term in a 3D inviscid vortex method. When the vorticity is convolved with the smoothing function, $\boldsymbol{\omega} \approx \zeta_\sigma * \boldsymbol{\omega}$, then the Laplacian can be approximated in this way: $\Delta \boldsymbol{\omega} \approx \Delta(\zeta_\sigma * \boldsymbol{\omega}) = \Delta \zeta_\sigma * \boldsymbol{\omega}$. Fishelov's method is obtained by approximating the convolution integrals by the trapezoid rule. The scheme is, like PSE, known to lose accuracy as the particles become disordered, and rezoning (see Section 6) has been used to counter this problem in conjunction with higher-order cutoff functions [38].

The diffusion velocity method is similar to a Lagrangian method that was used in plasma physics for collision kinetics [39], where a finite difference scheme in the moving mesh is implemented. This approach was modified as a particle method and also applied to the Fokker–Planck model in Reference [40]. In the context of vorticity dynamics, the method is obtained in Reference [41] by deriving a diffusion contribution to the velocity, which—in analogy to Fick's first law of diffusion—is proportional to the gradient of vorticity. It is also proportional to ν/ω, which encounters difficulties in areas where the vorticity is close to zero. It has been noted that diffusion is correctly modelled by this method only where particle overlap is maintained [42] so a nicely ordered particle field must somehow be provided; remeshing was used in Reference [43] in the context of the diffusion velocity method applied to a problem of natural convection.

Finally, there are a few variations of vortex methods based on triangulations, which find different ways to include viscosity. The 'free-Lagrangian' method [44] is based on a discrete approximation of the Laplacian operator on a triangulation of the particle positions, obtained from a Voronoi diagram. In Reference [44], the convection of vorticity was still solved with a particle representation (vortex blobs). A later inviscid implementation [45] abandons the blobs in favour of a scheme analogous to Lagrangian finite elements, where the vorticity is approximated by piecewise polynomials on a triangulation. In this case, the velocity is evaluated by a technique related to the fast multipole method, on the triangulated mesh.

In Reference [46] a method for obtaining the derivatives of vorticity based on a least-squares formulation is presented, which claims the advantage of preserving accuracy better for irregular points, in comparison with PSE or Fishelov's method. First, a quadratic polynomial function is fitted to control points in the neighbourhood of a particle location, and then derivatives of vorticity are obtained by explicit differentiation of the polynomials. The unknown coefficients in the polynomial expansion are found by minimizing errors in a least-squares sense, which results in a two-by-two linear system for each particle, giving the first and second derivatives at its location. If the particles were on a uniform grid and only near neighbours were used in the fit, the scheme would reduce to a standard centred difference. Applied to viscous flows, this technique is used in combination with a diffusion velocity implementation to provide the necessary expansion of the vorticity support. This 'moving least-squares' approach was demonstrated for axisymmetric flows in Reference [47], using diffusion velocity and calculating vorticity derivatives from the polynomial fit (recall that diffusion velocity requires knowledge of the vorticity gradient; second derivatives are also needed in the axisymmetric formulation). In 3D, these workers abandon the particle representation, and instead develop a 'tetrahedral vortex element' (TVE) method [48]. This scheme utilizes the moving least-squares method to obtain the derivatives in the vortex stretching and diffusion terms, and obtains the vorticity by interpolation on a tetrahedral mesh built from the Lagrangian points, similarly to the 2D triangulated vortex method of Reference [45]. The TVE method claims the advantages of providing a continuous interpolated vorticity field even when the Lagrangian points are highly disordered, or distributed anisotropically, and the fact that it is easy to avoid vorticity penetration into solid bodies (as there are no cutoff functions). Perhaps one could consider this a method closer to vortex-in-cell than to vortex blob methods, where the mesh is now unstructured. It is clearly not a grid-less method, as a tetrahedral mesh is constructed and fit to the Lagrangian points on each time step.

With so many different approaches to construct a viscous vortex method, it is clear that the field is still maturing and is yet far from a consensus in regards to the 'best' viscous scheme. Each method reviewed above has some desirable characteristics as well as some disadvantages. PSE is very sensitive to having an ordered distribution of particles, and so there has been a great amount of work on remeshing schemes. The method of superposing derivatives of the cutoff function, as used by Anderson and Greengard and Fishelov, is also dependent on a regular particle arrangement to maintain accuracy; diffusion velocity, as well, requires constant overlap of blobs. It would seem that the problem for an accurate viscous vortex method is not the viscous scheme, but spatial adaption. Indeed, to bypass the problem generated by irregular particle fields, the vortex redistribution method resorts to elaborate and computationally expensive algorithms, while the tetrahedral vortex element method does away with the particle representation altogether (at the cost of constructing a mesh at every time step). Even the core spreading method suffers from the problem of how to limit the size

that vortex blobs can grow, so needs a form of spatial adaption. But in contrast to the other methods, core spreading is utterly simple in its approach to satisfying diffusion effects. If the spatial accuracy can be maintained with some form of adaptive refinement, the core spreading method seems to offer the opportunity of a truly grid-less viscous vortex method.

4. NUMERICAL STUDY ON THE ACCURACY OF DISCRETIZATION

The first fundamental question one may pose in regards to the practical use of the vortex (blob) method refers to the accuracy that one can expect from the spatial discretization of the vorticity field, by means of Equation (2). Note that we do not deal with the point vortex method which presents certain difficulties due to its singular nature. Both the vortex blob method and the point vortex method have an extensive convergence analysis [49–55], but here we examine the accuracy of the blob approach in practice, based on numerical experiments.

In the first exploratory numerical calculations, we look at the errors obtained in the vorticity and velocity by discretizing a given vorticity field using vortex blobs. Two classic test problems were chosen to carry out this investigation, both problems of the simplest possible nature: an axisymmetric, inviscid vortex patch (16), and a Lamb–Oseen vortex (17):

$$\omega(r) = \begin{cases} (1-r^2)^3, & r \leqslant 1 \\ 0, & r > 1 \end{cases} \tag{16}$$

$$\omega(r,t) = \frac{\Gamma_0}{4\pi\nu t}\, \mathrm{e}^{-r^2/4\nu t} \tag{17}$$

where $r^2 = x^2 + y^2$. The first problem is particularly suited to observe the effects of features in the inviscid vortex method, as the exact solution consists of circular trajectories of different velocity and the initial particle distribution gets rapidly distorted due to the large shear (hence, this flow belongs to the class of problems known as 'circular shear layers'). The second problem is especially useful to consider different viscosity schemes, being the simplest viscous vortex flow and having an analytical solution; in addition, the vorticity transport equation reduces to the diffusion equation for this problem, so that the viscous effects are decoupled from the non-linear effects.

Since the study of Perlman [6], one of the early efforts to analyse and quantify the errors in vortex methods, it has become well-known that the accuracy of discretization with the vortex method fundamentally depends on three factors: the choice of cutoff function used in the discretization, the value of the cutoff parameter, σ, and the way an initially existing vorticity field is discretized and the numerical method initialized. Two choices of cutoff function were used in the present calculations: the most popular choice with most workers is the Gaussian (3), formally a second order cutoff, i.e. $O(\sigma^2)$; in addition, we used the algebraic 8th-order cutoff of Nordmark [38], given by

$$\zeta^{(8)}(r) = \begin{cases} \dfrac{52}{\pi}(1 - 21r^2 + 105r^4 - 140r^6)(1-r^2)^9, & r \leqslant 1 \\ 0, & r > 1 \end{cases} \tag{18}$$

with $\zeta_\sigma(\mathbf{x}) = 1/\sigma^d \zeta(|\mathbf{x}|/\sigma)$, d being the dimension of the problem.

Initialization of a vortex method calculation consists of determining the identifying quantities of the vortex particles, i.e. their location and circulation strength. Their core size is chosen initially as a discretization parameter, dictating the resolution of the calculation (the smallest scales that can be resolved). The initial particle locations are, for simplicity, most commonly chosen to be the nodes on a Cartesian mesh. An alternative would be to divide the support of the initial vorticity in cells of uniform size and initialize a particle in a random location inside the cell; this has been called 'quasi-random' initialization [25], while a random initialization will not construct any form of ordered lattice. On a square lattice of initial particle locations, one can assign the circulation strengths by simply evaluating the vorticity at the particle location and multiplying by the cell area (or volume in 3D). That is,

$$\Gamma_i = \omega_i^o h^d = \omega(t=0, \mathbf{x}_i) h^d \tag{19}$$

For the random or quasi-random initializations one could use again the local value of vorticity, and multiply by the average cell area (obtained by dividing the support of the vorticity by the number of particles, N). This approach is less accurate but may be preferable if the vorticity is not smooth; but the initialization on a square lattice and the use of (19) is the most commonly used initialization method. Note, however, that the use of (19) to initialize the particle strengths incurs an error, which is related to the midpoint rule; one can try to decrease this error by iterating on the Γ_i's, for example (as in Beale's method of circulation processing, see Section 6).

Indeed, the particle representation of the initial vorticity field incorporates a sort of numerical diffusion when (19) is used; this 'diffusive effect' of the blob discretization can be quantified for the particular case of Gaussian blobs used on flows which are *exact solutions* of the diffusion equation. This can be seen by writing the general solution of the heat equation as (in one dimension, for simplicity),

$$g(x,t) = \frac{1}{\sqrt{4\pi\nu t}} \int_{-\infty}^{\infty} g(x', t_o) \exp\left(-\frac{|x'-x|^2}{4\nu t}\right) \mathrm{d}x' \tag{20}$$

Writing the discrete representation of the integral above using single interval extrapolative rule, it can be seen that if one spatially discretizes the function $g(x,t)$ using Gaussian cores (with $k=2$) the sums obtained on the left- and right-hand sides of the equation can be made equal with $\sigma^2 = 2\nu t$. Hence, the discretization reconstructs the vorticity field after it has 'diffused' for a time interval of $\sigma^2/2\nu$. As an application, a scheme for improving the accuracy of initialization for test problems based on exact solutions of the diffusion equation—as in the Lamb–Oseen vortex—is the application of an initial 'anti-diffusion', equivalent to shifting the initial time by an amount $-\sigma^2/2\nu$. Hence, the initial circulation strengths for a Lamb–Oseen vortex at time-zero, t_o, are obtained by

$$\Gamma_i = \omega_i^o h^2 = \frac{\Gamma_0 h^2}{4\pi\nu(t_o - \sigma^2/2\nu)} \exp\left(-\frac{x_i^2 + y_i^2}{4\nu(t_o - \sigma^2/2\nu)}\right) \tag{21}$$

This 'time-shift correction' can be applied in the very particular situation of discretizing an exact solution of the diffusion equation using Gaussian blobs. As such, it will not be usable in a practical application; it allows, however, the production of accurate initialization of the Lamb vortex for the purposes of studying overlap dependence and the effects of the particle distribution. Overlap dependence is illustrated by the results plotted in Figure 1(a), where for

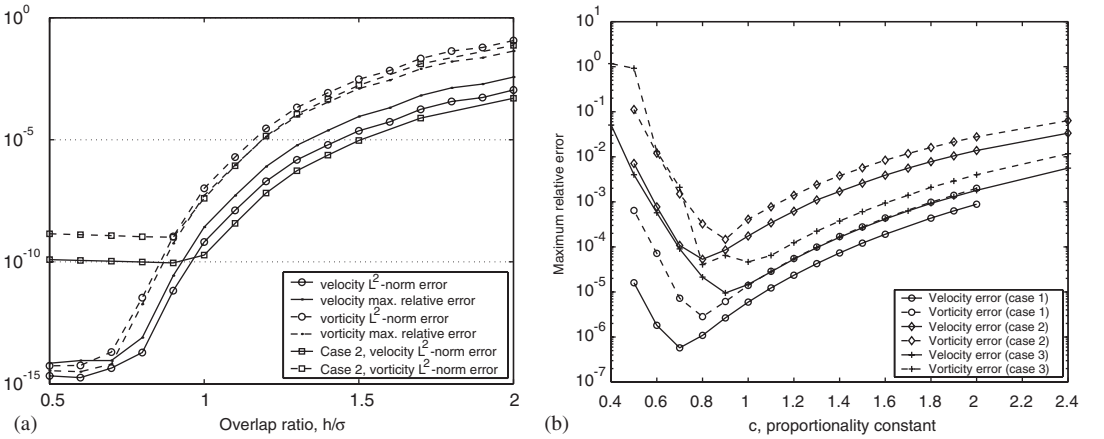

Figure 1. Lamb vortex, $\Gamma_o = 1$, $\nu = 0.01$: (a) Vorticity and velocity errors vs overlap ratio, using Gaussian blobs with $\sigma = 0.02$; and (b) vorticity and velocity errors vs c, with $\sigma = c\sqrt{h}$ for Nordmark blobs. Case 1: $t_o = 0.25$, $h = 0.01$, $N = 5177$; Case 2: $t_o = 0.25$, $h = 0.025$, $N = 925$; Case 3: $t_o = 0.5$, $h = 0.025$, $N = 925$.

a given choice of σ two initial conditions of the Lamb vortex are discretized with varying overlap ratio. Case 2 incorporates a criterion for blob 'population control' whereby after calculating particle strengths with (21), vortex blobs with a strength below 2.2×10^{-16} (equal to the machine roundoff in these experiments) were eliminated. As a result, the errors are bounded below by $O(10^{-10})$, but with significantly reduced problem size (about 40% less particles when $h/\sigma = 0.7$).

The error of the discretization was measured in terms of the maximum relative error in vorticity and velocity—i.e. the maximum field value of the point-by-point difference between the exact and discretized vorticity/velocity, divided by the maximum value of vorticity/velocity—and a discrete L^2-norm error, calculated using the following formula:

$$\|\omega - \omega^h\| = \left(\sum_i |\omega(\mathbf{x}_i) - \omega^h(\mathbf{x}_i)|^2 h^2 \right)^{1/2} \tag{22}$$

The results shown in Figure 1(a) demonstrate how, with the appropriate choice of the overlap ratio, it is possible in practice to obtain negligibly small errors with the vortex blob representation. In this case of the Lamb–Oseen vortex discretized with Gaussian blobs, an optimum overlap ratio in the range $(0.7, 1.0)$ will produce velocity errors smaller than about $O(10^{-8})$. It is clear as well how strongly the initialization depends on overlap ratio, there being a loss of several orders of magnitude in the accuracy as h/σ increases passing through the value of 1. Another observation that one can make from the figure is the fact that the vorticity errors are always larger than the velocity errors, which is consistent with the general results of the convergence theory of vortex methods.

When using high-order cutoff functions, it is typical to determine the initial overlap of blobs by the relation $\sigma = c\sqrt{h}$ and a choice of c; this follows Hald's convergence theory [50, 53]. Results of discretizing the Lamb vortex using (19) with Nordmark blobs and varying proportionality constant c are shown in Figure 1(b); minimal errors can be observed with

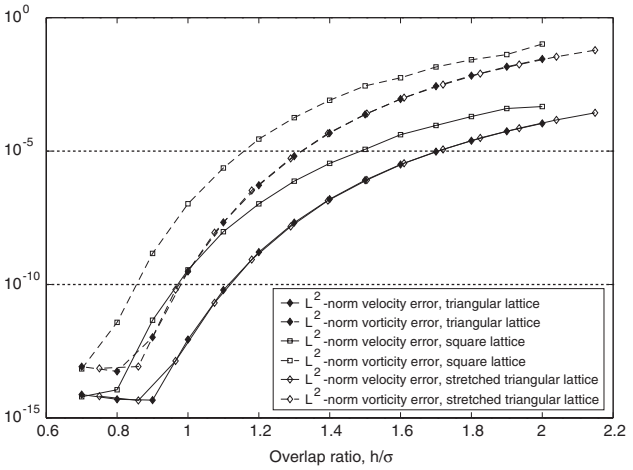

Figure 2. Lamb vortex $\Gamma_o = 1$, $t_o = 0.25$ and $\nu = 0.01$. Vorticity and velocity errors vs overlap ratio for Gaussian blobs with $\sigma = 0.02$, on square vs triangular lattice.

$c \in (0.7, 0.9)$. Note that standard initialization (19) is used with the Nordmark blobs, which provides a tolerable error because of the high-order of the cutoff function (the time shift correction may be applied only with Gaussian blobs).

The discretizations used for the plot on Figure 1(a) were carried out with particles placed on a square lattice. To observe the effect of a different particle arrangement, discretizations were carried out on the Lamb vortex with initial conditions as in case 1 of Figure 1(a), but using both a square lattice and a triangular lattice of equilateral triangles. This time, the discrete L^2-norm errors were measured on a finer mesh of spacing $h/2$ for h the inter-particle spacing. Discretizations on the triangular lattice were carried out twice: once with the same value of h as the ones on the square lattice, which produces in the denser triangular lattice a slightly increased resolution, and once with a 'stretched' value of h to obtain an equivalent resolution to the square lattice (that is, for equal cell area in both lattices). This last option results in an effective larger overlap ratio for the same 'nominal' value of h, for comparison with the same resolution case on a square lattice. For the same value of h, the triangular lattice 'fills space' 15% more in 2D, i.e. there are 15% more particles on average in the same area. With the stretched value of h, of course, the number of particles for each value of overlap ratio is almost the same for both lattices. The domain of initialization is obtained by first filling with blobs a square area considerably larger than the support of vorticity, then calculating the particle strengths using the time-shift correction and finally eliminating all particles whose calculated strength is less than the machine roundoff. The errors are calculated after this form of population control is enforced, which has the effect of limiting the accuracy that is obtained at the smallest values of overlap ratio.

Figure 2 shows how the triangular lattice provides an increased accuracy in comparison with a square lattice. For example, with an overlap ratio of 1.0 at the same value of h, the triangular lattice proves to be almost three orders of magnitude more accurate, although with a larger N (by about 15%, as already mentioned). If one compares the equivalent resolution cases for a nominal overlap ratio of 1.0, then the triangular lattice gives about one order of

magnitude improvement (with an effective overlap of 1.075). We will come back to this when discussing radial basis function interpolation, in Section 7.

5. ACCURACY ISSUES OF THE LAGRANGIAN FORMULATION

It was seen in the previous section how the accuracy of discretization with vortex blobs depends crucially on the overlap ratio. This demonstrates the practical situation when the vortex blobs cease to overlap fully (overlap ratio larger than 1, for Gaussian blobs). One expects, therefore, that as the vortex particles are allowed to evolve in a Lagrangian manner, the strain of the flow field will cause the overlap to be lost in some areas thereby increasing the errors. The problem becomes of crucial importance as the viscosity value is reduced, and it is the determining factor in the results that can be obtained from an inviscid simulation.

Consider a typical calculation of a Lamb vortex with small viscosity, giving a moderate Reynolds number of 2000. The initial vorticity is discretized using Gaussian blobs on a square lattice with the time-shift correction, resulting in an initial velocity error of $O(10^{-9})$ with $h/\sigma = 0.8$. As shown in Figure 3(a), the errors quickly grow, and oscillate around a deteriorated accuracy level 5 orders of magnitude larger than initially. Note that the errors do not grow without bound, which has been pointed out before [6–8]. Figure 3(b) shows the contours of vorticity error as well as particle locations at the final time of the calculation. The error contours show how the Lagrangian distortion of the particles introduces spurious (although weak) vortical structures. The vorticity error is measured in a maximum norm, and the contour of largest absolute value is 1.23% of the maximum vorticity, while the outermost contour has a value of 0.2% the maximum vorticity. In this calculation, vorticity diffusion was provided by core spreading and the final value of σ is 0.0490 (note that in a problem with radial symmetry, Greengard's inconsistency observations [22] do not apply).

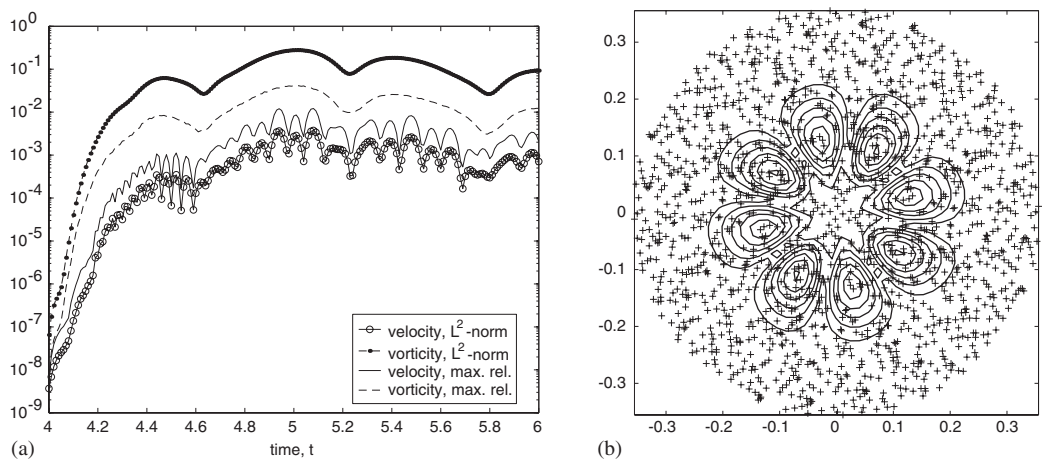

Figure 3. Calculation of a Lamb vortex: $\Gamma_o = 1$, $t_o = 4.0$ and $\nu = 0.0005$; Gaussian blobs with $\sigma = 0.02$, $h = 0.0160$, $N = 1560$. RK4 time-stepping, $\Delta t = 0.01$: (a) Evolution of errors in vorticity and velocity; and (b) vorticity error contours and particle locations at final time.

The loss of accuracy due to Lagrangian distortion of the particles is more dramatic when using high-order blob functions, which are more vulnerable to a reduced overlap. For this reason, in the early works that tested high-order blobs, 'rezoning' was immediately applied [7] (rezoning is explained in the next section). For example, discretizing the inviscid vortex patch with Gaussian and Nordmark blobs using the same value of h, a much more accurate result is obtained at time-zero with the latter, as expected. After only four time steps, however, the run with Nordmark blobs can show larger errors than the one with Gaussians. This was the case for calculations using $h=0.1$, resulting in $N=305$ in both cases, with $h/\sigma=0.5$ for the Gaussian and $c=1.6$ for the Nordmark blobs. With these parameters, the Nordmark blobs provided an improvement of 2–3 orders of magnitude in the time-zero errors, compared to the Gaussian blobs initialized with five iterations of circulation processing (Beale's method; see next section). Without spatial adaption, however, the Nordmark blobs result in larger errors by up to 2 orders of magnitude at the end of the run with 50 time steps.

This and the previous sections demonstrate that overlap control is important in vortex methods no matter which scheme is chosen to account for viscous effects. Some authors claim that an alternative to the PSE scheme that does not rely on quadrature rules will do away with the need for remeshing and therefore retain the grid-less nature of the vortex methods. In fact, no matter the viscous scheme, the accuracy of the vortex method depends on preserving overlap and some form of spatial adaption is compulsory to maintain the desired level of accuracy.

6. SPATIAL ADAPTION SCHEMES FOR VORTEX METHODS

6.1. Particle remeshing

Most vortex method application programs incorporate spatial adaptation in the form of 'remeshing' or 'rezoning' algorithms, which consist in re-starting the particle field on a regular grid every few time steps, and re-calculating the particle circulation strengths by interpolation or other means. This subject constitutes an important active area of research in vortex methods.

The prevailing approach consists in constructing a square lattice of new particle locations, and obtaining the circulation values on the nodes from the old particles by interpolation. The 2D or 3D interpolation rules are built by Cartesian tensor product of 1D kernels, and these have been constructed of increasing order in terms of the inter-particle separation, h. The commonly used interpolation kernels are of two families, the 'Λ' and the 'M' family. The first order Λ_0 and M_1 kernels are equivalent to nearest-grid-point (NGP) interpolation, and conserve only total circulation (they are almost never used in vortex methods). The so-called 'tent-function', Λ_1, is a second order interpolant and is equivalent to the M_2 kernel, conserving total circulation and linear impulse. Second order interpolants, however, are considered 'too dissipative' [30] and therefore the relevant formulas of the Λ family for current vortex methods are the following:

$$\Lambda_2=\begin{cases}1-u^2 & \text{if } 0\leqslant u\leqslant \frac{1}{2},\\ \frac{1}{2}(1-u)(2-u) & \text{if } \frac{1}{2}\leqslant u\leqslant \frac{3}{2},\\ 0 & \text{otherwise}\end{cases}\qquad \Lambda_3=\begin{cases}\frac{1}{2}(1-u^2)(2-u) & \text{if } 0\leqslant u\leqslant 1\\ \frac{1}{6}(1-u)(2-u)(3-u) & \text{if } 1\leqslant u\leqslant 2\\ 0 & \text{otherwise}\end{cases} \tag{23}$$

The third-order Λ_2 kernel corresponds to quadratic interpolation and it conserves up to second moment of vorticity, with a 3^d stencil. This scheme has been used successfully in bluff body flow computations [26, 56]. Its main disadvantage is lack of smoothness, as it is not even continuous. The piecewise-cubic and continuous kernel Λ_3 (Everett's 4th-order formula) requires a 4^d stencil, conserving up to third moment of vorticity. (Indeed, the Λ family of kernels is constructed by specifying that increasing moments of vorticity be conserved.)

The interpolation kernels of the 'M' family are derived from splines; they are characterized by being more regular than the Λ family. Central B-splines are capable of interpolating exactly only linear functions, however, so they cannot be of higher than second order. The improved M_4' kernel

$$M_4' = \begin{cases} 1 - \frac{5}{2}u^2 + \frac{3}{2}u^3 & \text{if } 0 \leqslant u \leqslant 1 \\ \frac{1}{2}(1-u)(2-u)^2 & \text{if } 1 \leqslant u \leqslant 2 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

introduced in Reference [57] is of higher accuracy (3rd order) and when used in vortex methods will conserve the first three invariants (total circulation, linear impulse and angular impulse). Researchers have used M_4' remeshing when they seek highly accurate results, e.g. Reference [28]; it has the advantage of being quite smooth (class C^1) and it was shown in Reference [30] to be considerably more accurate than the Λ_2 scheme which is of the same formal order.

Using one of the interpolation kernels above, the remeshing schemes obtain the contribution of circulation $\Delta\Gamma_{j,i}$ from the i-th old vortex with Γ_i to the new mesh point $(\tilde{x}_j, \tilde{y}_j)$ according to (in the two-dimensional case):

$$\Delta\Gamma_{j,i} = \Gamma_i \Lambda\left(\frac{\tilde{x}_j - x_i}{h}\right)\Lambda\left(\frac{\tilde{y}_j - y_i}{h}\right) \tag{25}$$

where Λ represents the 1D interpolation kernel.

6.2. *Rezoning*

What has been termed 'rezoning strategy' [7] is similar to remeshing, in the sense that the particle field is re-started on a rectangular mesh of new vortex blobs. But instead of interpolating the circulation of the old particles onto the new (neighbouring) ones, the rezoning strategy consists in evaluating the vorticity induced by the old particles at the new blob locations, and then obtaining the new particles' circulation by applying the standard initialization formula (19). In other words, one defines a continuous vorticity field using the current particle distribution, and then evaluates this function on the new blob locations $\tilde{\mathbf{x}}$, i.e. summing over all j current blobs:

$$\tilde{\omega}(\tilde{\mathbf{x}}, t) = \sum_j \Gamma_j \zeta_\sigma(\tilde{\mathbf{x}} - \mathbf{x}_j) \tag{26}$$

To evaluate this function at a new blob location on a square mesh, one has to add the vorticity contribution of each old particle, requiring $O(N \cdot M)$ operations for N old blobs and M new ones. The scheme is therefore expensive computationally, similar to the direct summation of the Biot–Savart law (although in theory one could implement some form of

fast summation; the literature does not to our knowledge show any such implementation in a rezoning scheme). A criterion for determining how often to execute a rezoning process was introduced in Reference [38], but it is based on calculating at every time step an estimate of the 'vorticity error along vortex paths', which is just as expensive. On the other hand, note that if one obtains the new circulation strengths using the calculated vorticity values at the new blob locations and Equation (19), that is, multiplying the new blob's vorticity by cell area h^2, then an error equivalent to the initialization error is incurred. For this reason, it seems reasonable to advocate that rezoning as described by Beale and Majda should be applied only when using high-order blobs.

6.3. Numerical experiments with remeshing and rezoning

Numerical experiments show that classic rezoning is effective in controlling errors due to Lagrangian distortion when using high-order blobs. In addition, as it is exemplified by the calculations in Figure 4, when using the 8th order cutoff classic rezoning gives higher accuracy than standard remeshing with the M_4' kernel. Figure 4 shows errors obtained in calculations of the inviscid vortex patch, where the Nordmark blobs give at this rather low resolution (only 305 blobs with $h = 0.1$) an initial L^2-norm velocity error of 3.05×10^{-6}. Without spatial adaption, after only five time steps the error has grown to 4.71×10^{-4}, and it is 1.35×10^{-2} at the end of the run. Remeshing with M_4' every 2 time steps (the velocity error is 1.2×10^{-5} at $t = 2\Delta t$, so one may prefer to adapt every time step), the final velocity error is 8.01×10^{-4}, whereas using rezoning the final error is 2.96×10^{-4}. But in the vorticity field the difference is more substantial: final L^2-norm error of 2.86×10^{-2} with remeshing vs 2.18×10^{-3} with rezoning. It is interesting to note that in the case using M_4' remeshing, most of the accuracy loss occurs on the first remesh; this 'jump' is what we call the initial remesh error. Another issue to consider is that in the 25 remeshing processes of this run, the problem size grew about 7 times to $N = 2148$, due to the fact that the interpolation rule requires a rather wide stencil and new vortices must each time be added all around the vorticity support. With rezoning, there is no reason in theory for the number of particles to grow (in practice, it may grow a little due to programming simplifications).

The situation seen in the middle plot of Figure 4, where the first remeshing event produces a jump in the errors, is characteristic of many numerical experiments performed in this work.

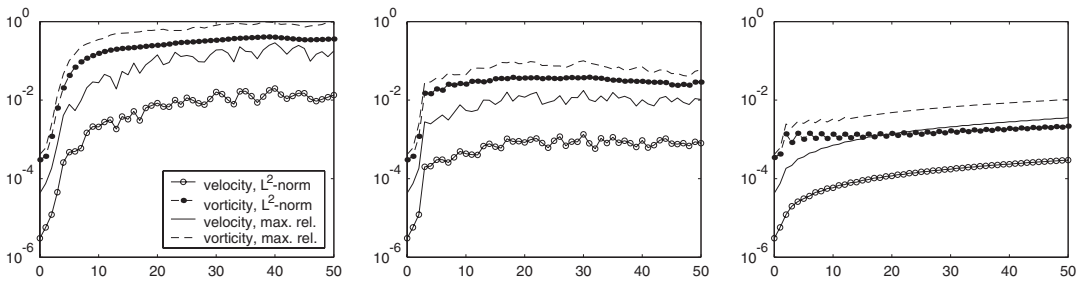

Figure 4. Vorticity and velocity errors vs time for calculations of the inviscid vortex patch. Nordmark blobs with: $h = 0.1$, $c = 1.6$ and $\sigma = c\sqrt{h} = 0.51$, $N = 305$. Time stepping: RK4, $\Delta t = 1$. Left: no spatial adaption; middle: remeshing with M_4' every 2 time steps; right: classic rezoning every 2 time steps.

Many times, the initial remesh error is as much as two or three orders of magnitude; on occasions, even more. Note, however, that this only becomes apparent when having a very accurate initialization, either using the 8th order cutoff, or with Gaussians using the time-shift correction in the Lamb vortex case or Beale's method of circulation processing (explained next) in the case of the inviscid vortex patch. Several experiments shown in the literature utilizing standard initialization with Equation (19) have much larger initial errors, and so the 'jump' in the first remeshing is not visible. For instance, the calculations of the inviscid vortex patch (16) in Reference [30] show for $h=0.05$ an initial L^2 norm error of about 4×10^{-4} (reading from the plot), which deteriorates in 100 time steps to 4×10^{-2} without any remeshing. With M_4' remeshing carried out at every time step, the final error improves to about 2×10^{-3}, and no error jump is visible at any stage. This paper unfortunately does not specify what type of cutoff was used (we suspect that it may be part of the thesis work of Reference [58], in which case a 4th order algebraic cutoff was used). Using the simple Gaussian blobs and the same value of $h=0.05$, we initialize the vortex patch with circulation processing obtaining a velocity error of 6×10^{-7}; applying M_4' remeshing every five time steps, the error after 100 steps is 3×10^{-4}. But most significantly, an initial remesh error is observed which brings the velocity error immediately to order 10^{-5} at $t=2\Delta t$. In the experiment of Reference [30] the initial error was already one order of magnitude larger than this, and no jump can be seen.

6.4. *Circulation processing*

An alternative approach to control the effects of Lagrangian distortion, not based on relocating the blobs on a regular lattice, is processing the circulation values to improve the approximation of the discretized vorticity field to the exact one at the particle locations. The idea of recalculating the circulations can be understood as a way of adjusting the particle volumes to account for the changes in overlap. To improve on the approximation given by (2) at a given time, one may search for new values γ_i of the circulation strengths so that

$$\sum_{i=1}^{N}\gamma_i\zeta_\sigma(\mathbf{x}_j(t)-\mathbf{x}_i(t))=\omega(\mathbf{x}_j,t)=\omega_j \tag{27}$$

where ω_j is the vorticity at the particle location, which one wishes to approximate better. Multiplying this equation by h^2, the area of a blob in a regular, square mesh, one obtains

$$h^2\sum_{i=1}^{N}\gamma_i\zeta_\sigma(\mathbf{x}_j(t)-\mathbf{x}_i(t))=\Gamma_j \tag{28}$$

where the Γ_j's are the known, current circulation values of the blobs. Equation (28) represents a linear system for the coefficients γ_i, which can be written in matrix form as

$$A\boldsymbol{\gamma}=\boldsymbol{\Gamma} \tag{29}$$

where, $A_{ij}=h^2\zeta_\sigma(\mathbf{x}_j(t)-\mathbf{x}_i(t))$. An iterative method to solve this system was proposed by Beale [8], who observed that the previous circulation values are a natural first guess for the γ_i's. First rewriting the system (29) as

$$(A-I)\boldsymbol{\gamma}+\boldsymbol{\gamma}=\boldsymbol{\Gamma} \tag{30}$$

then Beale's iterative method is expressed by the following rule:

$$\gamma_i^{n+1} = \Gamma_i + \gamma_i^n - \sum_j h^2 \gamma_j^n \zeta_\sigma(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \tag{31}$$

This iterative method was used at every time step in Reference [8], as well as in Reference [9] in combination with the PSE viscous scheme. However, Beale's method is not guaranteed to converge, and in the example given in p. 209 of Reference [25] one can see that even though an improvement of two orders of magnitude in the velocity accuracy is obtained at time-zero, the errors do seem to grow persistently so that the final accuracy is just slightly better than the unprocessed case.

6.5. Vortex blob splitting

Blob splitting was proposed as a means to rectify the lack of consistency in the bare core spreading method [23], by maintaining the maximum blob radius below a stipulated value, $\sigma_{\max}$. The method consists in replacing a single vortex blob of width $\sigma > \sigma_{\max}$ with a number (chosen as 4) of children blobs of width $\alpha\sigma$. The algorithm is controlled by the numerical parameter α, while other parameters are determined by imposing the conservation of vorticity moments. Since the splitting generates exponential growth of the problem size, a merging algorithm is proposed to alleviate this problem. A convergence analysis is developed in Reference [23] which proves linear convergence to the Navier–Stokes equations in the L^∞ norm. (As is pointed out in the paper, the classic convergence theory of Beale and Majda [51] utilizes the L^p norm in the full non-linear case, and also the standard convergence theory is applied to the velocity, while Rossi utilizes the vorticity instead.) The error of the refinement is proved to converge to zero as $\alpha \to 1$, and for the particular choice of 1:4 refinement it is proved to be bounded by $O(1-\alpha^2)$ in the linear case.

The basic resolution parameter in the splitting algorithm is the fixed maximum blob width, $\sigma_{\max}$. The accuracy parameter of the splitting is α, determining the width $\alpha\sigma$ of the children blobs. The children blobs' circulation is of course $\frac{1}{4}$ the strength of the parent, for conservation of zeroth moment of vorticity (total circulation). They are placed with their centre at a distance r from the parent blob's centre, where r is determined from conservation of second moment of vorticity, giving $r = \sqrt{2}\sigma(1-\alpha^2)^{1/2}$ in the case of blobs with cutoff (3). In Reference [23], the values of α are chosen between 0.7 and 0.9 (with a different definition of the Gaussian cutoff), and numerical experiments are presented using a Lamb vortex and a co-rotating vortex pair, where this latter case was compared with an RVM calculation using much larger N. In the case of the Lamb vortex, plotted results of tangential velocity at a given location show quite visible errors, while the plots showing blob locations reveal that there is loss of overlap in some areas. In the case of the co-rotating vortices, the figures also show overlap loss, in particular near the edges of each vortex. The calculation is said to compare well with RVM, but it must be said that at the Reynolds number of the experiment, equal to 100, random walk can be quite crude. It would seem, then, that the chosen splitting scheme results in rather low accuracy.

One problem with the splitting method is the lack of any overlap control. In addition, the scheme introduces numerical diffusion. Furthermore, it is not clear that the criterion for calculating r based on conservation of second moment of vorticity is the best choice. An alternative scheme was studied in [59], where r is chosen to conserve the value of vorticity at the parent blob's centre. In one-dimensional experiments, it was found that the centre

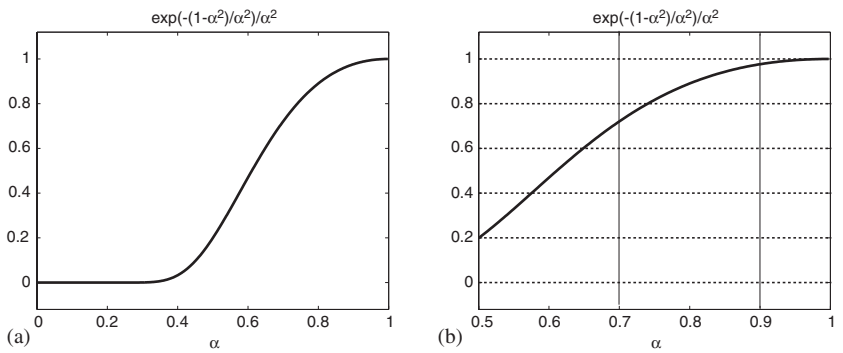

Figure 5. Ratio of the maximum vorticity of the four children blobs to the maximum vorticity of the parent blob placed at the origin for Rossi's 1:4 splitting, resulting in $1/\alpha^2 \exp(-(1-\alpha^2)/\alpha^2)$. Zoom to usual α values in (b).

vorticity constraint is more accurate than conservation of second moment of vorticity for one-blob splitting, but it was less accurate in an experiment with eleven blobs superimposed. Since neither constraint proved to be 'ideal', a set of two-dimensional tests using a Lamb vortex were performed with the aim of finding an empirical relationship $r = f(\alpha)$. A linear fit was performed to (α, r) pairs obtained on the basis of minimizing L^1-norm errors of vorticity, comparing with the analytical solution, which resulted in a relationship somewhere in between the centre vorticity and second moment constraints. This empirical law was built into the code used for the core spreading experiments in Reference [60], one of the few recent implementations of the core spreading vortex method.

For the case of splitting with the second moment criterion, the diffusive character of the method is illustrated in Figure 5. The plots are obtained from the ratio between the centre vorticity after splitting and before splitting one blob, and one can see that the result is always less than 1 (i.e. the maximum vorticity always decreases after splitting).

7. HIGHER ACCURACY WITH MESHLESS SPATIAL ADAPTION

7.1. Radial basis function interpolation

The problem of spatial adaption for vortex methods—i.e. determining the identifying quantities (location and circulation) of a new set of well-overlapped particles to best approximate the current vorticity field—can be approached as a problem of function approximation. In this perspective, one can see a similarity between the vortex blob discretization given by (2) and the technique of radial basis function (RBF) interpolation, a tool for solving multivariate scattered data interpolation problems. In the context of Beale's method of circulation processing, the problem of finding the new circulation values so that the particle representation best approximates the vorticity field has been recognized as a problem of 'scattered data interpolation' in References [8, 9]. They both refer to Franke's review paper [61], where a study is made of different algorithms for the problem of scattered data interpolation on a set of known surfaces. Franke assessed about 30 algorithms and ranked them based on six criteria—

accuracy, visual aspect, sensitivity to parameters, execution time, storage requirements and ease of implementation. The two methods that ranked best, the so-called multi-quadrics (MQ) and thin-plate splines (TPS) are examples of radial basis functions. But since Franke's work, a significant volume of literature has been published on this subject.

The problem of scattered data interpolation is that of how to approximate an unknown function $f \in C(\Omega)$ whose values are known on a set $X = \{x_1, \ldots, x_N\} \subset \Omega \subset \mathbb{R}^d$. The RBF approach, where we follow the notation of [62], is to choose the function that approximates f to be of the form

$$s_{f,X}(x) = \sum_{j=1}^{N} \alpha_j \Phi_j(x, x_j) + p(x) \tag{32}$$

where $p(x)$ is a low-degree polynomial, and $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a fixed function that is translation invariant and in particular satisfies

$$\Phi(x, y) = \phi(\|x - y\|_2), \quad \text{with } \phi : [0, \infty) \to \mathbb{R} \text{ (radiality)} \tag{33}$$

Clearly, the blob discretization of the vorticity is analogous to the interpolant (32), where the polynomial part is chosen as null and the basis function is the cutoff function, a Gaussian for example. Indeed, Gaussians are used in RBF interpolation (and usually with null polynomial part), as are basis functions of the following types:

$$\text{(i) } \phi(r) = r^{\beta}, \ \beta > 0, \ \beta \notin 2\mathbb{N} : \ \text{'pseudo-cubics'} \tag{34}$$

$$\text{(ii) } \phi(r) = r^{2k} \log(r), \ k \in \mathbb{N} : \ \text{'thin-plate splines'} \tag{35}$$

$$\text{(iii) } \phi(r) = (c^2 + r^2)^{\beta}, \ \beta > 0, \ \beta \notin \mathbb{N} : \ \text{'multi-quadrics'} \tag{36}$$

The solution of (32) requires the satisfaction of the interpolation conditions by collocation, leading to a linear system for the coefficients $\alpha = (\alpha_1, \ldots, \alpha_N)$ and the polynomial coefficients. For our purposes, we can assume that the polynomial part is null, and write the system as

$$\mathbf{\Phi}\alpha = \vec{f} \tag{37}$$

where $\vec{f}$ represents the vector of function values at the centres, $\vec{f} = \{f(x_1), \ldots, f(x_N)\}$, and $\mathbf{\Phi}_{ij} = \phi(\|x_i - x_j\|)$. The matrix $\mathbf{\Phi}$ being full and not well conditioned, Franke [61] concluded that global basis function methods are not feasible for large N. But since then, a great deal of work has contributed to effectively resolve this and several other difficulties. Preconditioning operators were first introduced in Reference [63] for the cases of the MQ and TPS, based on triangulation of the data points and construction of discrete approximations to the iterated Laplacian operators, Δ^k. At the same time, progress was made in regards to theoretical aspects of the problem, with the result of Reference [64] that the interpolation system (37) is guaranteed a solution whenever the function $\Phi(x, y)$ is strictly conditionally positive definite and the data distinct. As described in Reference [62] (where proofs are given), the theory has been greatly extended and many basis functions have been characterized, for example, the MQ interpolant is conditionally positive definite and can be made positive definite by appending a linear polynomial, and the Gaussian is positive definite hence not requiring a polynomial part.

The problem of ill-conditioning of the distance matrix $\mathbf{\Phi}$ has been subject of extensive analyses aiming to establish bounds on the norm of the inverse, $\|\mathbf{\Phi}^{-1}\|$, and on the spectral condition number of $\mathbf{\Phi}$ for different basis functions. In References [65, 66] upper bounds on the inverse, for a given basis function, are found to depend only on the dimension of the domain space $\mathbb{R}^d$ and the *separation radius* of the data locations, which is defined by

$$q_X := \frac{1}{2} \min_{1 \leqslant j \neq k \leqslant N} \|x_j - x_k\| \tag{38}$$

This density measure of a point set is the minimal (half) distance that separates one site to its nearest neighbour in the data set. If it is very small, it means that two data locations are very close together, which in turn makes the distance matrix close to singular. The estimates on the condition number, $\kappa(\mathbf{\Phi}) = \|\mathbf{\Phi}\|\|\mathbf{\Phi}\|^{-1}$, depend additionally on N, through the dependence of $\|\mathbf{\Phi}\|$ on N. For strictly positive definite basis functions (including the Gaussian), however, estimates independent of N were given in Reference [67]. These estimates reveal that the condition number becomes very large as the minimal overlap ratio, which we could define as $q_X/(2\sigma)$ for the Gaussian, becomes small. This article also demonstrates how, for the Gaussian, the requirement of good conditioning is at odds with the accuracy of the interpolation, what is called the problem of 'good conditioning vs good fit'. Additional results in regards to upper bounds on the inverse $\|\mathbf{\Phi}^{-1}\|$ are found in Reference [68], whereas Reference [69] provides lower bounds for special cases of regular arrangements of the data, and [70] provides general lower bounds for scattered data which are not expressed as a function (explicitly) of separation radius. In this last work, it is shown how the lower bounds for $\|\mathbf{\Phi}^{-1}\|$ depend on the smoothness of Φ, becoming larger as the smoothness increases; also, support is provided to the idea that regular placement of the data is most favourable to the conditioning.

Much theoretical progress has been made, in addition, in the estimation of the error of interpolation with RBF's. Error bounds of arbitrarily high order were proved for the multi-quadrics in References [71, 72], and for Gaussians in Reference [73], where a simpler theoretical approach is used. Both of these works require a certain restrictive condition on the Fourier transform of the function f being approximated. The error estimates refer to the pointwise difference between the function f and the interpolant $s_{f,X}$, and are found to be of $O(h_X^k)$ where k depends on the RBF $\mathbf{\Phi}$ and the density measure h_X is called the *fill distance* and defined by

$$h_{X,\Omega} := \sup_{x \in \Omega} \min_{1 \leqslant k \leqslant N} \|x - x_k\| \tag{39}$$

The fill distance measures the maximal distance from any point $x \in \Omega$ (not necessarily a data location) to its nearest point in the data set. In the terminology of computational geometry, it is the radius of the largest empty circle in the data. Hence, it measures how the data 'fills' the support Ω, and the quality of the approximation using RBFs (for all different choices of RBF) will deteriorate as $h_{X,\Omega}$ gets larger. For some basis functions, including the multi-quadrics and the Gaussian, improved error estimates based on $h \leqslant h_{X,\Omega}$ were found in Reference [74]; these local estimates are $O(\lambda^{1/h})$ as $h \to 0$, with $0 < \lambda < 1$. Subsequently, a representation of the norm of the error functional for different RBFs that is workable numerically has been developed

[75, 76]. This representation is bounded by the so-called *power function*, $P(x)$, in the following manner:

$$|f(x) - s_f(x)| \leqslant |f|_{F_\Phi} \cdot P(x) \tag{40}$$

where F_Φ is an inner-product space of functions defined via Φ and $|\cdot|_{F_\Phi}$ is the seminorm defined in the function space. For the Gaussian, F_Φ is the space of C^∞ functions. The upper bound of $P(x)$ is written in the form $P^2(x) \leqslant F(h(x))$ where $h(x)$ is a local measure of fill distance. For example, the Gaussian $\Phi(x) = \mathrm{e}^{-\beta\|x\|^2}$ has $F(x) = \mathrm{e}^{-\delta/h^2}$, with $\delta > 0$. Note that this most favourable case of exponential convergence for Gaussians comes at the cost of the worst situation in terms of conditioning, due to it being very smooth. This was developed into a general notion of an *uncertainty principle* for RBF interpolation by Schaback in Reference [76], which says that good reproduction quality is only obtained at the cost of poor numerical stability and *vice versa*. By introducing increased smoothness constraints on the function f, approximation orders are doubled in Reference [77], but for the Gaussian with exponential error bounds, this does not bear a major improvement.

Finally, the extensive research in RBF interpolation has also made significant progress in regards to computational efficiency. First, consider that evaluating a function whose approximation has been expressed as an expansion in RBFs can be quite expensive, involving $O(N)$ operations for each evaluation point. Multipole expansions for the fast evaluation of the interpolant (32) were introduced for the TPS in Reference [78], where in addition the fast algorithms are also applied to the matrix–vector product required at each step of an iterative solution method, in particular the pre-conditioned conjugate-gradient method. A new method for the fast evaluation of RBF expansions, based on generalizing the multipole method so that changes of basis are easily performed, was presented in Reference [79]. Second, the actual solution of the RBF interpolation problem can be prohibitive for large N, unless fast methods are implemented. This was successfully addressed in [80], where preconditioning strategies were used in conjunction with fast matrix–vector multiplication and a GMRES iterative solution method. The preconditioning method is based on changing the basis in which the RBFs are represented, using approximate cardinal functions. Numerical experiments with TPS and MQ demonstrated significant clustering of the eigenvalues, improving the condition number by several orders of magnitude. This lead to the GMRES solution converging in only a few iterations. The overall strategy entails $O(N \log N)$ operations and $O(N)$ storage. Alternatively, an approach for the fast solution of RBF interpolation analogous to forward substitution is developed in References [81, 82], based on generalization of the iterative method constructed in Reference [83]; this was applied to TPS while the extension to the MQ and other conditionally positive definite functions was shown to be accessible. Considerable improvements to this method have been performed, in particular the inclusion of a Krylov subspace algorithm which is guaranteed to converge [84]. A third approach for the efficient solution of the RBF interpolation system is based on domain decomposition [85]; such a method was applied to data sets of up to 5 million points in a two-dimensional application.

7.2. Comment: significance and application in vortex methods research

The previous section amounts to a compressed review of the advancement in the field of radial basis function interpolation. Clearly, this field comprises a substantial body of knowledge, untapped by vortex methods workers. There is a great deal of cross-over applications that one

can think of, for example in regards to refinements of the convergence theory of the vortex blob method. For example, most convergence proofs start by assuming standard initialization using (19), whereas it is apparent that a much increased accuracy can be obtained by solving the RBF interpolation system by collocation. But first of all, we find that the ideas of RBF interpolation permit the development of fully meshless vortex methods, where spatial adaption can be provided to overcome the problem of Lagrangian distortion of the particle field, and also to correct the core spreading method. Spatial adaption based on the RBF formulation consists of restarting the particle field onto a well-overlapped but not necessarily regular lattice, thus allowing for easily conforming to boundaries. In contrast, a square lattice is obligatory for both the rezoning strategy and any remeshing scheme based on tensor products. Moreover, the vortex blob discretization uses the core size, σ, as a parameter determining the spatial resolution; it is necessary that the core size be small to keep convection errors small. Sometimes, it may be desirable to have different core sizes in different parts of the domain, for example one could allow a larger error in the far wake of bluff body flows. Different core sizes cannot be dealt with directly in the usual remeshing approach based on exchanging circulation; instead, some workers have devised schemes based on carrying out the remeshing on a mapped domain [31, 86–88], thus allowing for spatially varying cores. If one views the problem of spatial adaption as one of approximating accurately the vorticity like an RBF interpolation problem, there is no obstacle to redefining the core sizes at the same time that one carries out the spatial adaption. On the one hand, one can define variable resolution in the physical domain, and on the other one can counter the problem of having cores that are too large when the core spreading method is used to account for viscous effects. That is, the RBF adaptation procedure can easily restart the particles to a smaller core radius. The next section will provide demonstrations of these ideas in numerical experiments, which in addition evidence that higher accuracy can be obtained in comparison with standard remeshing. It would seem that a grid-less formulation and high accuracy are essentially connected.

In a more advanced application, one can envisage that the error estimates obtained for RBF interpolation, developed to the point where there are numerically workable bounds, could be used to design truly adaptive spatial refinement. By this we allude to the meaning generally ascribed to the term *adaptive* in other sub-disciplines of CFD, such as finite elements and panel methods. The common meaning is that the scheme should perform a certain measurement of an error indicator, and use it to derive a refinement of the discretization (e.g. an unstructured mesh, in the finite element context). In addition, an adaptive method should act only locally, but with the objective of reducing the error globally in the domain. For example, having both of these attributes, locality and being measurement based, one can view Rossi's splitting to be indeed an adaptive refinement method, as he calls it. Although, being based on measurement of blob size, instead of an actual error indicator, it is a simpler and more crude form of adaptivity. In particular, the splitting cannot guarantee a given accuracy (of the global vortex method calculation), due to the fact that it does not control blob overlap.

The notion of using local error measurement to provide adaptive refinement using the RBF ideas has not been implemented as yet, but some simple numerical experiments with an approach akin to rezoning have been performed. Instead of using (19) to obtain the new circulations, of course, they are solved for by collocation (see next section for details). But an RBF adaption scheme with the feature of acting only locally can be realized, for example, based on ideas such as coarsening and refinement using the Voronoi diagram, as developed in Reference [89]. The node adaption strategy proposed in this work consists of using an error

indicator which is function of the node set to assign a significance value to each node, which in turn is used to flag nodes to be refined or coarsened. Coarsening is performed by removing the node itself, and refinement is performed by the insertion of its Voronoi points. Thus, this approach is fully local, adaptive, and grid-less.

7.3. Numerical experiments with RBF spatial adaption of vortex blobs

Presently, numerical experiments will be presented with a radial basis function formulation of the spatial adaption process, and comparisons will be made with the standard remeshing schemes. A number of experiments were performed in which the particle locations were restarted every few time steps, and the circulation values of the new blobs were obtained by solving an RBF interpolation problem. The new locations were chosen on regular lattices, square and triangular, but only for convenience as the formulation does not require any regular particle arrangement.

Consider first a series of calculations with Gaussian blobs on a Lamb vortex with $\Gamma_o = 1.0$, $t_o = 0.2$, $\nu = 0.001$; the value of initial core radius is $\sigma = 0.03$ and diffusion is provided by core spreading. Figure 6 presents the evolution of errors for three cases: no spatial adaption, remeshing with M_4' every 10 time steps, and RBF adaption every 10 time steps. The initialization was performed on a square lattice using the time-shift correction, which gives very small errors, but the time evolution cannot preserve this (effects of time stepping, convection error and loss of the uniform lattice); upon the first time step the accuracy is reasonably good with an L^2-norm velocity error of 5.6×10^{-7}. It can be seen that without spatial adaption the errors start growing quite dramatically after about 10 time steps, and by the end of the calculation the velocity error is 3.5×10^{-4} (with an intermediate maximum value of 1.8×10^{-3}). Adding M_4' remeshing, the final velocity error is only slightly smaller, at 2.5×10^{-4}, but the maximum values of errors during the whole calculation are about 70% smaller for velocity and 50% smaller for vorticity (both error measures, L^2 norm and maximum norm). The initial remesh error is obvious, in this case a jump of almost two orders of magnitude. The final value of core size for these two runs is $\sigma = 0.07$. The remeshing calculation was repeated but using instead the Λ_3 kernel (not shown), with very similar results, but a slightly smaller initial remesh error. The plot on the right of Figure 6 shows the results obtained when the vortex blobs were restarted on the initial square grid, and their circulation values were obtained by solving the

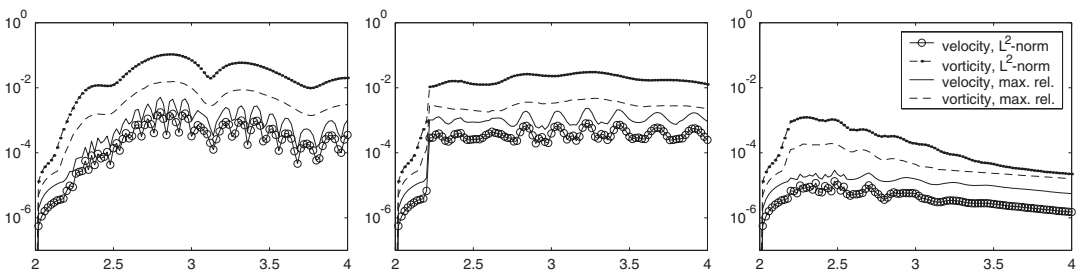

Figure 6. Vorticity and velocity errors vs time for calculations of the Lamb vortex. Gaussian blobs with: $\sigma = 0.03$, $h/\sigma = 0.8$ and $h = 0.024$, $N = 1153$. Time stepping: RK4, $\Delta t = 0.02$. Left: no spatial adaption; middle: remeshing with M_4' every 10 time steps; right: RBF adaption (solved with SOR) every 10 time steps.

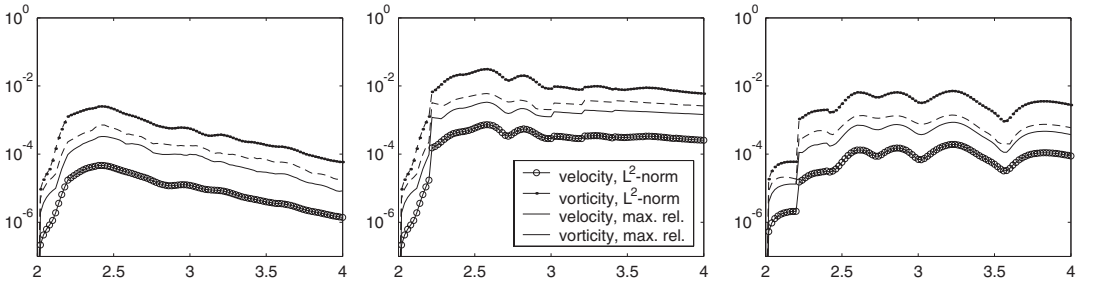

Figure 7. Vorticity and velocity errors vs time; Lamb vortex, Gaussian blobs: $\sigma = 0.03$, $h/\sigma = 0.8$; nominal $h = 0.024$ in a triangular lattice. RK4, $\Delta t = 0.02$. Left: RBF adaption every 10 time steps; middle: hexagonal redistribution scheme every 10 time steps; right: hexagonal redistribution with halved value of h.

RBF-type linear system using SOR with underrelaxation. This solution scheme was chosen due to it having been used before to solve for the circulations upon initializing an elliptical vortex patch in [28]. Note that with the RBF adaption scheme there is not a visible error upon the first processing event, demonstrating an immediate and considerable improvement over standard remeshing. The final L^2-norm velocity error in this case is 1.5×10^{-6}, consisting of a two-order-of-magnitude improvement over remeshing. Finally, and importantly, in this calculation the core sizes were restarted to their initial values upon each RBF adaption event, so that the maximum value of σ was 0.036. In other words, the correction for the core spreading method is provided in addition to accurate spatial adaption by location processing.

Similar experiments were performed using an initial particle distribution on a lattice of equilateral triangles. This time, both for the initialization and the RBF adaption processes, the circulations were obtained using a GMRES scheme developed for RBF interpolation of geophysical data in Reference [90]. See Figure 7, left. The L^2-norm velocity error is 2.16×10^{-7} upon the first time step, and the final value is 1.4×10^{-6}. All errors are slightly smaller than the case on the square lattice, even though the resolutions were made equivalent by 'stretching' the value of h here, to obtain the same cell area. At this point, we will digress somewhat to discuss this increased accuracy in the triangular lattice, which was pointed out previously in the end of Section 4. According to our discussion of RBF interpolation, in particular in regards to the error estimates, it is expected that the errors be smaller for particle arrangements with smaller fill distance. In a square lattice, the radius of the largest empty circle is equal to $\sqrt{2}/2 \cdot h = 0.707h$. In an equivalent resolution triangular lattice, equal cell area gives: $\sqrt{3}/2 \cdot h_t^2 = h_s^2$, where h_s is the point separation in the equivalent square lattice. The radius of the largest empty circle in the triangular lattice is $\sqrt{3}/3 \cdot h_t$, which combined with the previous relationship gives $0.621h_s$; i.e. the triangular lattice 'fills space' better, and the RBF interpolation is more accurate.

For comparison with remeshing on the triangular lattice, a new hexagonal redistribution scheme developed at GALCIT was used; this scheme is a higher (third-) order version of the face-centred schemes introduced in Reference [91]. Its advantages over the M_4' scheme are a narrower stencil, so it injects less particles on the edges of the domain, and a more isotropic distribution of circulation. Figure 7, centre, shows the errors obtained with this remeshing scheme under the same parameter combination as the calculation with RBF adaption, left. The result is quite similar to previously, when M_4' was compared with the RBF adaption

case on the square lattice (where SOR was used to solve for the circulations); there is a visible initial remesh error and the final L^2-norm velocity error is 2.5×10^{-4}. All errors are very similar to the previous case with M_4' remeshing (Figure 6, centre), but one can see that the error curves are more 'flat'; this is the effect of the improved isotropy characteristics of the hexagonal redistribution. In Figure 7, right, a calculation is shown where the value of h was halved. This results in an improvement, but it does not equal the accuracy of the RBF adaption case. It was found that the value of h has to be between 2 and 3 times smaller to obtain the same accuracy with standard remeshing, at this resolution. This means that in 2D the problem size can be up to an order of magnitude smaller when one uses the RBF formulation, to obtain the same accuracy as with remeshing.

Similar results as those described above were obtained on the inviscid test problem. For example, discretizing the vortex patch (16) using Gaussian blobs on a square lattice, initialized with five iterations of circulation processing (Beale's method), with $\sigma = 0.05$, $h/\sigma = 1.0$, $N = 1245$, and $\Delta t = 0.02$ (RK4), an initial L^2-norm velocity error of 5.6×10^{-7} is obtained. The final velocity error after 60 time steps without remeshing is 1.3×10^{-4}, while with M_4' remeshing every 10 steps the final error is reduced to 6.5×10^{-5}. Most of this error is suffered upon the initial remeshing process, however. Note that this initial remesh error would not have been visible if circulation processing at time-zero had not been performed, which increased the initial velocity accuracy by almost 4 orders of magnitude. For this reason, other numerical experiments with this test problem reported in the literature, which start with less accurate initializations, describe very good results with M_4' remeshing and do not observe the initial remesh error; for example, References [30, 25, p. 235]. A calculation of the inviscid vortex patch with the same parameters, but using RBF adaption instead of remeshing, was carried out to a final time about three times larger than above, obtaining a final velocity error of 4.2×10^{-6}. This calculation used SOR to solve for the circulation strengths. More details and error plots of the experiments with the inviscid vortex described in this paragraph were presented in Reference [92].

The numerical experiments described in this section are very encouraging. Indeed, several more experiments were performed which support the conclusion that the vortex method has the potential for an increased accuracy. The problem seems to be that the standard remeshing approach brings back the mesh to an otherwise meshless method, thereby adding numerical dissipation. Evidently this is more important when calculating higher Reynolds number flows. The advantages of the RBF adaption are increased accuracy, in addition to a truly meshless formulation. The first advantage can be more dramatic at high resolution; e.g. one experiment with the Lamb vortex using $h = 0.0175$ produced a final velocity error of 3×10^{-8}, which provided a four-order of magnitude improvement over remeshing; see this result in Reference [92]. Many more numerical experiments and additional details, including a grid-refinement study, can be found in Reference [93]. Finally, the approach allows one to correctly use the core spreading method, thereby avoiding the errors and difficulties associated with other means of accounting for viscous effects.

8. PROOF-OF-CONCEPT CALCULATION: A RELAXING PERTURBED MONOPOLE

When a strong vortex is subject to a perturbation, the first fundamental question that arises is whether the vortex will return to an axisymmetric shape. Also, one is interested in the time

scale of the relaxation process. In the case of an $m=2$ perturbation (quadrupole), the main effect is a localized elliptical deformation of the main vortex. The question of axisymmetrization of elliptical vortices has been subject of much interest, initiated by Melander *et al.* [94] who argued that relaxation towards axisymmetry is a generic process of two-dimensional vortices.

An axisymmetric Gaussian monopole with a quadrupolar perturbation was studied numerically using the vortex method with core spreading and splitting/merging in Reference [10]. It was found that the perturbed monopole axisymmetrizes for small perturbations, but relaxes to a tripolar attractor if the perturbation is strong enough. When this is the case, the positive portion of the perturbation is homogenized through mixing, while the negative part is persistent, thereby forming a quasi-steady rotating tripole. Rossi's numerical experiments were performed in a range of Reynolds numbers between 10^3 and 10^4, and erosion of the tripole structure was seen for the smaller Reynolds number.

Presently, we have computed this same problem using the vortex method with core spreading and RBF adaption, and compare results with [10]. The initial condition is the superposition of the axisymmetric vortex ω_o, and a perturbation ω':

$$\omega_o(\mathbf{x}) = \frac{1}{4\pi}\,\mathrm{e}^{-|\mathbf{x}|^2/4}, \quad \omega'(\mathbf{x}) = \frac{\delta}{4\pi}\,|\mathbf{x}|^2\mathrm{e}^{-|\mathbf{x}|^2/4}\,\cos 2\theta \tag{41}$$

where $\theta = \arg(\mathbf{x})$ and δ is the strength of the perturbation; Rossi used a value of $\delta = 0.25$ and observed the quasi-steady tripole. With this value of the perturbation and $Re = 10^4$, we have performed a calculation using Gaussian blobs with $\sigma = 0.2$, $h/\sigma = 0.9$ resulting in $N \cong 4600$ (unfortunately, Rossi does not list any numerical parameters used). RBF adaption was performed every 10 time steps, resulting in a maximum core size $\sigma_{\max} = 0.205$. The results are presented in Figure 8 in the form of contour plots of total vorticity and vorticity perturbation, which can be compared with Figure 1 of [10] (reproduced here in part, with permission). It can be seen that the results compare very well, in terms of the shape of the contours and the rotation angle of the structure, but the results of Reference [10] look quite a bit more rugged. It is suspected that this is the effect of the numerical errors of the splitting/merging processes. This is supported by results of calculations with different Reynolds numbers, at the same time. Figure 9 presents the contours of vorticity perturbation at $t = 500$ for the three values of Re: 10^3, 5×10^3, 10^4. Comparing with the results of Reference [10] (Figure 8: reproduced here with permission), it can be seen that for the two higher Reynolds number calculations, the shape of the structure and its angular position correspond very closely with Rossi's calculations, where once again the contours of the present calculation look smoother. However, for the lowest Reynolds number Rossi observed a much more pronounced erosion of the tripole structure (note that in Figure 8 of Reference [10], left, the negative and positive contours are swapped). This difference may be attributed to the numerical errors in the splitting/merging. Indeed, in a calculation with a lower Reynolds number of 500 we still did not see the vorticity erosion that Rossi experiences for $Re = 10^3$. These calculations support the assertion that RBF adaption can provide an increased accuracy, while at the same time correcting the core spreading viscous method.

The quasi-steady nature of the tripole attractor means that it evolves slowly due to viscous effects on the Reynolds number timescale. This decay is illustrated in Figure 10(a), where the additional case of $Re = 500$ has been included. This plot shows the ratio of minimum to maximum vorticity vs Reynolds timescale, where the decay is similar at different Reynolds

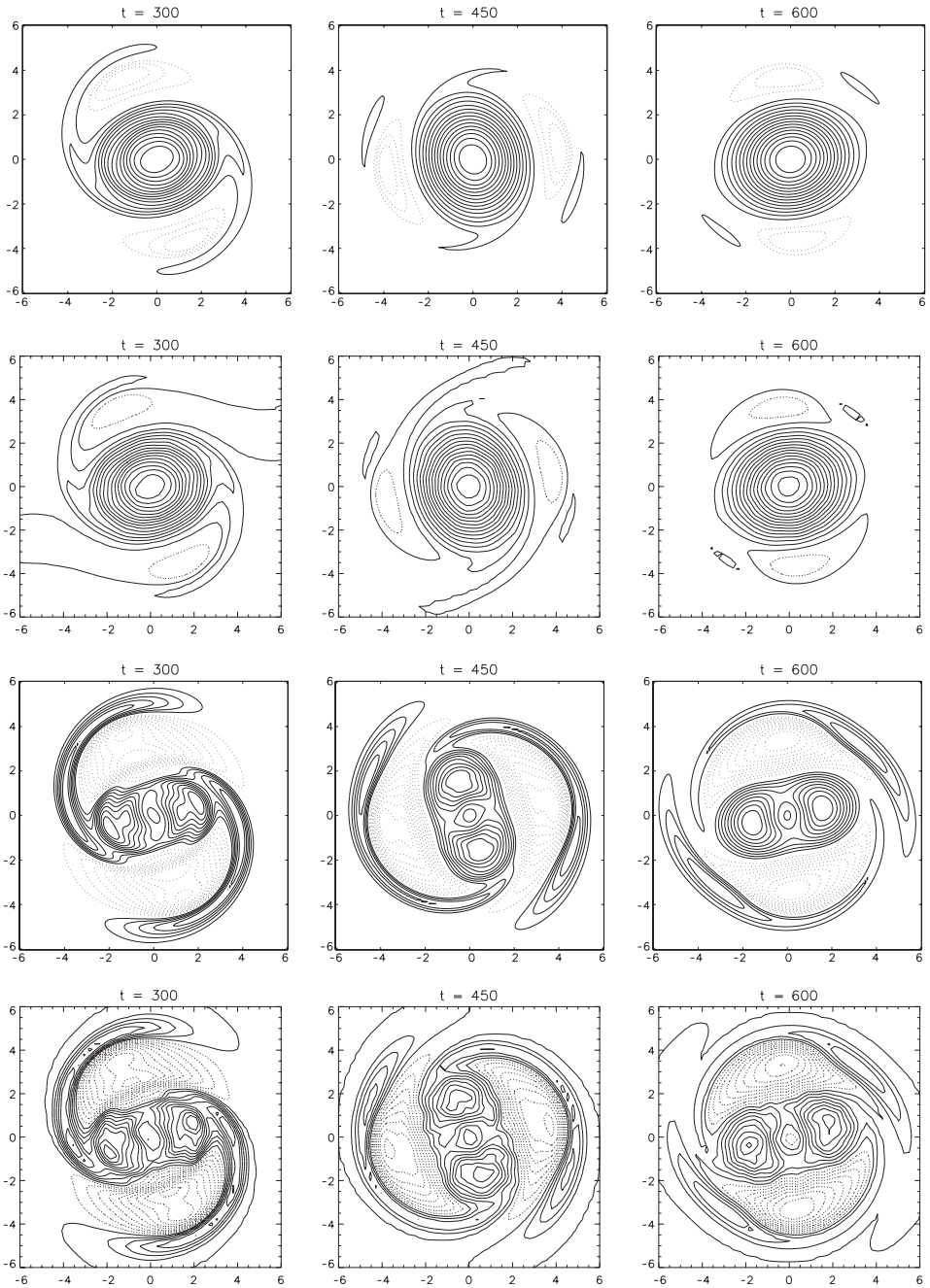

Figure 8. Perturbed monopole relaxing to a tripole attractor. $Re = 10^4$, $\delta = 0.25$. Top two rows: total vorticity field; bottom two rows: perturbation vorticity. Negative contours in dotted line. Rows 1 and 3: present method. Rows 2 and 4: Rossi's vortex blob splitting method, reproduced from Reference [10] with permission.

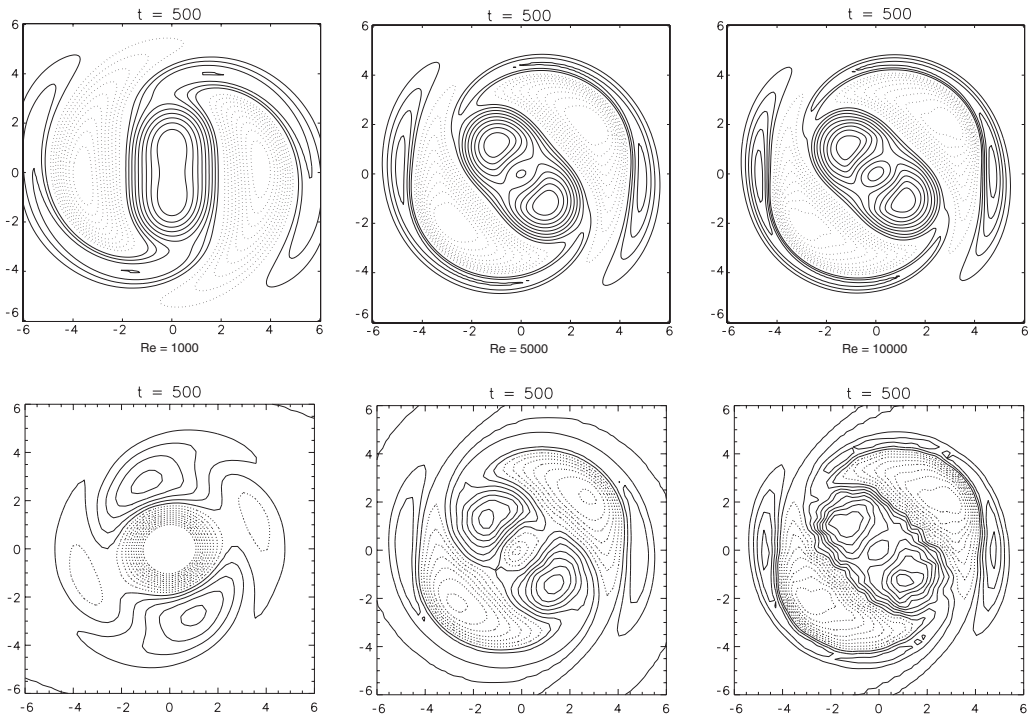

Figure 9. Perturbed monopole relaxing to a tripole attractor; perturbation vorticity at varying Reynolds number: from left to right, $Re = 1000$, 5000, 10^4. First row: present method, maximum vortex blob core size, respectively, 0.245, 0.21, 0.205. Second row: Rossi's splitting method, reproduced from Reference [10] with permission.

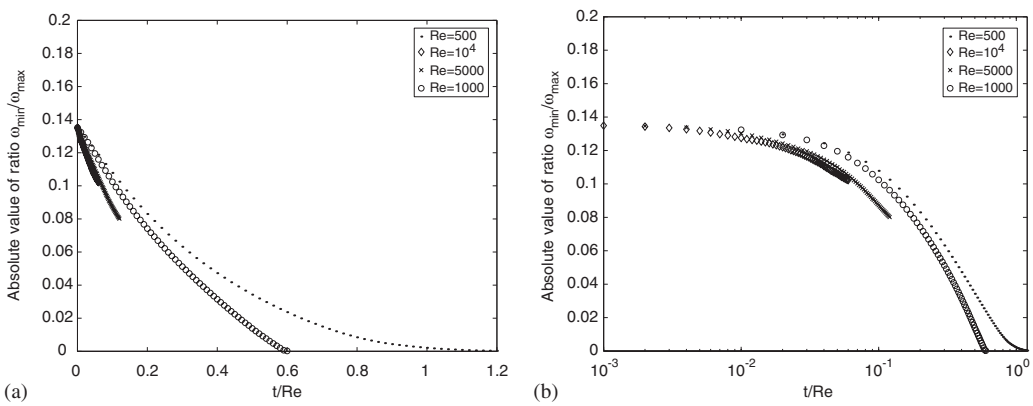

Figure 10. Decay of tripole structure on the Reynolds number time scale. Ratio of minimum to maximum vorticity values vs t/Re on a linear scale (a); and a logarithmic scale (b).

numbers. The present calculations, however, seem to show a more marked retardation of the decay for the lower *Re* in comparison with Reference [10]. Figure 10(b) shows the same data plotted with a logarithmic scale on the abscissa, which helps to see the early re-organization of the vortical structure. For all Reynolds numbers, the inclusions of negative vorticity tend to vanish and the asymptotic state is an axisymmetric vortex.

9. CONCLUDING REMARKS

This article has presented a vortex method using a formulation based on radial basis function (RBF) interpolation for the spatial adaption of the Lagrangian vortices, which preserves the grid-free nature of the method. The method has been tested with various numerical experiments, and we conclude that there is a link between a meshless formulation and an increased accuracy. Numerical experiments show considerably more accuracy than standard remeshing, and also better results than blob splitting for the core spreading method.

The paper also has ingredients of a review in the areas of: viscous schemes for vortex methods, spatial adaption schemes, and RBF interpolation. The topic of viscous schemes has been discussed with the aim of presenting a case for the core spreading vortex method. It is argued that a difficulty for viscous vortex schemes is the Lagrangian nature of the method and the need for spatial adaption. As is well known, the widespread use of the particle strength exchange (PSE) method has promoted the development of remeshing schemes, the avoidance of which has in turn motivated the development of quite complicated or computationally expensive methods (vortex redistribution method, VRM; tetrahedral vortex element, TVE). Other alternatives like Fishelov's method and diffusion velocity also require very nearly uniform particle distributions. On the other hand, the core spreading method suffered mathematical objections, which were recently corrected using a spatial adaption scheme (blob splitting) that unfortunately is not very accurate—comparable to random walk, or RVM. Other than that, core spreading shares with RVM the features of being local, simple, parallelizable and grid-free—with the addition of being deterministic.

The research on RBF interpolation has been reviewed to show that the considerable progress in this area provides many avenues for cross-over applications in vortex methods. An RBF-based adaption scheme allows for a truly meshless method, with opportunities for developing a truly adaptive spatial refinement (local, based on defined criteria: error, stability), the provision of correction for core spreading that's more accurate than blob splitting, and permitting implementation of variable resolution in the physical domain (this constitutes research in progress). Furthermore, advanced fast methods of solution are available.

Other interesting observations made here include the fact that in the choice of cutoff function one should consider that higher-order cutoffs are more vulnerable to loss of overlap. Also, the optimality of a triangular lattice of vortices, in terms of providing maximum accuracy of discretization and possibly the most important observation is the fact that the standard remeshing schemes limit the accuracy of the vortex method, as even with a very accurate initialization one consistently observes an 'initial remesh error'. Experiments with RBF adaption show an immediate improvement by the absence of this error. These last observations (omitting the first) are consistent with remarks made quite early, as in Reference [5] one reads: '... the most significant numerical error arises from the anisotropic [cloud-in-cell,] CIC-interpolation

of velocities ... The only way to [improve this] is either to adopt a more complex interpolation algorithm which produces an isotropic velocity field from a point vortex (r-dependence only) or to employ a different mesh structure (for example a hexagonal or triangular mesh).' Standard remeshing as well as classic rezoning both require square particle lattices, while the RBF formulation does not. It rests to investigate the degree to which the accuracy is reduced if non-optimal scattered blob locations were used. Comparing results of RBF adaption vs standard remeshing on square and triangular lattices, however, the numerical experiments performed show improvements in accuracy of between one and four orders of magnitude, the most considerable improvements corresponding to the higher-resolution calculations (consistent with a higher convergence rate for RBF interpolation). In a given resolution, it was found that about an order of magnitude less particles could be used to obtain the same accuracy with RBF adaption as with remeshing (in 2D). For this reason, one can be optimistic on the application of these methods for high Reynolds number calculations.

ACKNOWLEDGEMENTS

Computations were carried out in the Iris Lab computers of the Graduate Aeronautical Laboratories, Caltech. Thanks are due to S. Billings for making available his GMRES code for RBF interpolation.

REFERENCES

1. Chan WM, Gomez RJ (III), Rogers SE, Buning PG. Best practices in overset grid generation. *AIAA #2002-3191; 32nd Fluid Dynamics Conference*, St. Louis MI, 2002.
2. Praeger W. Die Druckverteilung an Körpern in ebener Potentialströmung. *Physikalische Zeitschrift* 1928; **XXIX**:865–869.
3. Rosenhead L. The formation of vortices from a surface of discontinuity. *Proceedings of the Royal Society of London Series A* 1931; **A134**(823):170–192.
4. Greengard L, Rokhlin V. A fast algorithm for particle simulations. *Journal of Computational Physics* 1987; **73**:325–348.
5. Christiansen JP. Numerical simulation of hydrodynamics by the method of point vortices. *Journal of Computational Physics* 1973; **13**:363–379.
6. Perlman M. On the accuracy of vortex methods. *Journal of Computational Physics* 1985; **59**:200–223.
7. Beale JT, Majda A. High order accurate vortex methods with explicit velocity kernels. *Journal of Computational Physics* 1985; **58**:188–208.
8. Beale JT. On the accuracy of vortex methods at large time. In *Computational Fluid Dynamics and Reacting Gas Flows*, Engquist B *et al.* (eds). Springer: New York, 1988; 19–32.
9. Choquin JP, Lucquin-Desreux B. Accuracy of a deterministic particle method for Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1988; **8**:1439–1458.
10. Rossi LF, Lingevitch JF, Bernoff AJ. Quasi-steady monopole and tripole attractors for relaxing vortices. *Physics of Fluids* 1997; **9**:2329–2338.
11. Chorin AJ. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics* 1973; **57**:785–796.
12. Leonard A. Vortex methods for flow simulation. *Journal of Computational Physics* 1980; **37**:289–335.
13. Leonard A. Computing three-dimensional incompressible flows with vortex elements. *Annual Review of Fluid Mechanics* 1985; **17**:523–559.
14. Spalart PR. Vortex methods for separated flows. *NASA Technical Memorandum #100068*, 1988.
15. Sarpkaya T. Computational methods with vortices. *Journal of Fluids Engineering* 1989; **11**:5–52.
16. Puckett EG. Vortex methods: an introduction and survey of selected research topics. In *Incompressible Computational Fluid Dynamics: Trends and Advances*, Gunzburger MD, Nicolaides RA (eds). Cambridge University Press: Cambridge, 1993; 335–408.
17. Milinazzo F, Saffman PG. The calculation of large Reynolds number two-dimensional flow using discrete vortices with random walk. *Journal of Computational Physics* 1977; **23**:380–392.
18. Goodman J. Convergence of the random vortex method. *Communications in Pure and Applied Mathematics* 1987; **40**:189–220.
19. Long DG. Convergence of the random vortex method in two dimensions. *Journal of the American Mathematical Society* 1988; **1**:779–804.

20. Kuwahara K, Takami H. Numerical studies of two-dimensional vortex motion by a system of points. *Journal of the Physical Society of Japan* 1973; **34**:247–253.
21. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, 1967.
22. Greengard C. The core spreading vortex method approximates the wrong equation. *Journal of Computational Physics* 1985; **61**:345–348.
23. Rossi LF. Resurrecting core spreading vortex methods: a new scheme that is both deterministic and convergent. *SIAM Journal on Scientific Computing* 1996; **17**:370–397.
24. Degond P, Mas-Gallic S. The weighted particle method for convection–diffusion equations. Part 1. The case of an isotropic viscosity. *Mathematics of Computation* 1989; **53**:485–507.
25. Cottet GH, Koumoutsakos P. *Vortex Methods. Theory and Practice*. Cambridge University Press: Cambridge, 2000.
26. Koumoutsakos P. Simulation of unsteady separated flows using vortex methods. *Ph.D. Thesis*, California Institute of Technology, 1993.
27. Koumoutsakos P, Leonard A. High resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics* 1995; **296**:1–38.
28. Koumoutsakos P. Inviscid axisymmetrization of an elliptical vortex. *Journal of Computational Physics* 1997; **138**:821–857.
29. Leonard A, Shiels D, Salmon JK, Winckelmans GS, Ploumhans P. Recent advances in high resolution vortex methods for incompressible flows. *AIAA #97-2108*, 1997.
30. Cottet GH, Ould Salihi ML, El Hamraoui M. Multi-purpose regridding in vortex methods. In *ESAIM Proceedings*, *III International Workshop on Vortex Flows and Related Numerical Methods*. Toulouse, France, 24–27 August 1998, vol. 7. European Series in Applied and Industrial Mathematics, Giovannini A *et al.* (eds). Société de Mathématiques Appliqués et Industrielles, 1999; 94–103.
31. Ploumhans P, Winckelmans GS. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *Journal of Computational Physics* 2000; **165**:354–406.
32. Ploumhans P, Winckelmans GS, Salmon JK, Leonard A, Warren MS. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at $Re = 300,500$ and 1000. *Journal of Computational Physics* 2002; **178**:427–463.
33. Poncet P. Vanishing of mode B in the wake behind a rotationally oscillating circular cylinder. *Physics of Fluids* 2002; **14**:2021–2023.
34. Cottet GH, Poncet P. Particle methods for direct numerical simulations of three-dimensional wakes. *Journal of Turbulence* 2002; **3**:038.
35. Shankar S, van Dommelen L. A new diffusion procedure for vortex methods. *Journal of Computational Physics* 1996; **127**:88–109.
36. Fishelov D. A new vortex scheme for viscous flows. *Journal of Computational Physics* 1990; **86**:211–224.
37. Anderson C, Greengard C. On vortex methods. *SIAM Journal on Numerical Analysis* 1985; **22**:413–440.
38. Nordmark H. Rezoning for higher order vortex methods. *Journal of Computational Physics* 1991; **97**:366–397.
39. Fronteau J, Combis P. A Lie-admissible method of integration of Fokker–Planck equations with non-linear coefficients (exact and numerical solutions). *Hadronic Journal* 1984; **7**:911–930.
40. Degond P, Mustieles FJ. A deterministic approximation of diffusion equations using particles. *SIAM Journal on Scientific and Statistical Computing* 1990; **11**:293–310.
41. Ogami Y, Akamatsu T. Viscous flow simulation using the discrete vortex model—the diffusion velocity method. *Computers and Fluids* 1991; **19**:433–441.
42. Clarke NR, Tutty OR. Construction and validation of a discrete vortex method for the two-dimensional incompressible Navier–Stokes equations. *Computers and Fluids* 1994; **23**:751–783.
43. Ogami Y. Simulation of heat-vortex interaction by the diffusion velocity method. In *ESAIM Proceedings*, *III International Workshop on Vortex Flows and Related Numerical Methods*, Toulouse, France, 24–27 August 1998, vol. 7. European Series in Applied and Industrial Mathematics, Giovannini A *et al.* (eds). Société de Mathématiques Appliqués et Industrielles, 1999; 314–324.
44. Russo G. A deterministic vortex method for the Navier–Stokes equations. *Journal of Computational Physics* 1993; **108**:84–94.
45. Russo G, Strain JA. Fast triangulated vortex methods for the 2D Euler equations. *Journal of Computational Physics* 1994; **111**:291–323.
46. Marshall JS, Grant JR. A Lagrangian collocation method for vorticity transport in viscous fluid flows. In *Proceedings of the Forum on Vortex Methods for Engineering Applications*, Sandia National Laboratories, Albuquerque, NM, 1995.
47. Marshall JS, Grant JR. A Lagrangian vorticity collocation method for viscous, axisymmetric flows with and without swirl. *Journal of Computational Physics* 1997; **138**:302–330.
48. Marshall JS, Grant JR, Gossler AA, Huyer SA. Vorticity transport on a Lagrangian tetrahedral mesh. *Journal of Computational Physics* 2000; **161**:85–113.
49. Hald O, Del Prete VM. Convergence of vortex methods for Euler's equations. *Mathematics of Computation* 1978; **32**:791–801.

50. Hald O. Convergence of vortex methods for Euler's equations II. *SIAM Journal on Numerical Analysis* 1979; **16**:726–755.
51. Beale JT, Majda A. Vortex methods I: convergence in three dimensions. *Mathematics of Computation* 1982; **39**:1–27.
52. Beale JT, Majda A. Vortex methods II: high order accuracy in two and three dimensions. *Mathematics of Computation* 1982; **39**:29–52.
53. Hald O. Convergence of vortex methods for Euler's equations III. *SIAM Journal on Numerical Analysis* 1987; **24**:538–582.
54. Cottet GH, Goodman J, Hou TY. Convergence of the grid-free point vortex method for the three-dimensional Euler equations. *SIAM Journal on Numerical Analysis* 1991; **28**(2):291–307.
55. Hou TY, Lowengrub J, Krasny R. Convergence of a point vortex method for vortex sheets. *SIAM Journal on Numerical Analysis* 1991; **28**(2):308–320.
56. Koumoutsakos P, Shiels D. Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate. *Journal of Fluid Mechanics* 1996; **328**:177–227.
57. Monaghan JJ. Extrapolating B-splines for interpolation. *Journal of Computational Physics* 1985; **60**:253–262.
58. Ould Salihi ML. Couplage de méthodes numériques en simulation directe d'écoulements incompressibles. *Ph.D. Thesis*, Université Joseph Fourier, 1998.
59. Shiels D. Simulation of controlled bluff body flow with a viscous vortex method. *Ph.D. Thesis*, California Institute of Technology, 1998.
60. Shiels D, Leonard A. Investigation of a drag reduction on a circular cylinder in rotary oscillation. *Journal of Fluid Mechanics* 2001; **431**:297–322.
61. Franke R. Scattered data interpolation: tests of some methods. *Mathematics of Computation* 1982; **38**(157): 181–200.
62. Schaback R, Wendland H. Characterization and construction of radial basis functions. In *Multivariate Approximation and Applications*, Dyn N, Leviatan D, Levin D, Pinkus A (eds). Cambridge University Press: Cambridge, 2001; 1–24.
63. Dyn N, Levin D, Rippa S. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**(2):639–659.
64. Micchelli CA. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation* 1986; **2**:11–22.
65. Narcowich FJ, Ward JD. Norm of inverses and condition numbers for matrices associated with scattered data. *Journal of Approximation Theory* 1991; **64**:69–94.
66. Narcowich FJ, Ward JD. Norm estimates for the inverses of a general class of scattered data radial function interpolation matrices. *Journal of Approximation Theory* 1992; **69**:84–109.
67. Narcowich FJ, Sivakumar N, Ward JD. On condition numbers associated with radial-function interpolation. *Journal of Mathematical Analysis and Applications* 1994; **186**:457–485.
68. Ball K. Eigenvalues of Euclidean distance matrices. *Journal of Approximation Theory* 1992; **68**:74–82.
69. Ball K, Sivakumar N, Ward JD. On the sensitivity of radial basis interpolation to minimal distance separation. *Journal of Approximation Theory* 1992; **68**:401–426.
70. Schaback R. Lower bounds for norms of inverses of interpolation matrices for radial basis functions. *Journal of Approximation Theory* 1994; **79**:287–306.
71. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions. *Constructive Approximation* 1988; **4**:77–89.
72. Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions: II. *Mathematics of Computation* 1990; **54**(189):211–230.
73. Wu ZM, Schaback R. Local error estimates for radial basis function interpolation of scattered data. *IMA Journal of Numerical Analysis* 1993; **13**:13–27.
74. Madych WR, Nelson SA. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory* 1992; **70**:94–114.
75. Schaback R. Comparison of radial basis function interpolants. In *Multivariate Approximation. From CAGD to Wavelets*, Jetter K, Utreras F (eds). World Scientific: London, 1993; 293–305.
76. Schaback R. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics* 1995; **3**:251–264.
77. Schaback R. Improved error bounds for scattered data interpolation by radial basis functions. *Mathematics of Computation* 1999; **68**(225):201–216.
78. Beatson RK, Newsam GN. Fast evaluation of radial basis functions: I. *Computers and Mathematics with Applications* 1992; **24**(12):7–19.
79. Beatson RK, Newsam GN. Fast evaluation of radial basis functions: moment-based methods. *SIAM Journal on Scientific Computing* 1998; **19**(5):1428–1449.
80. Beatson RK, Cherrie JB, Mouat CT. Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics* 1999; **11**:253–270.

81. Powell MJD. A new iterative method for thin plate spline interpolation in two dimensions. *Annals of Numerical Mathematics* 1997; **4**:519–527.
82. Faul AC, Powell MJD. Proof of convergence of an iterative technique for thin plate spline interpolation in two dimensions. *Advances in Computational Mathematics* 1999; **11**(2–3):183–192.
83. Beatson RK, Goodsell G, Powell MJD. On multigrid techniques for thin plate spline interpolation in two dimensions. In *The Mathematics of Numerical Analysis*, Renegar J, Shub M, Smale S (eds). American Mathematical Society: Rhode Island, 1995; 77–97.
84. Faul AC, Goodsell G, Powell MJD. A Krylov subspace algorithm for multiquadric interpolation in many dimensions. *DAMTP 2002/NA05*, University of Cambridge, April 2002.
85. Beatson RK, Light WA, Billings S. Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM Journal on Scientific Computing* 2000; **22**(5):1717–1740.
86. El Hamraoui M. Contributions à la simulation d'écoulements tridimensionnels par méthode de vortex. *Ph.D. Thesis*, Université Paul Sabatier, 1999.
87. Cottet GH, Koumoutsakos P, Ould Salihi ML. Vortex methods with spatially varying cores. *Journal of Computational Physics* 2000; **162**:164–185.
88. Ploumhans P. Simulation of high Reynolds number flows past bluff bodies using vortex and boundary element methods. *Ph.D. Thesis*, Universtité Catholique de Louvain, 2001.
89. Behrens J, Iske A, Pöhn S. Effective node adaption for grid-free semi-Lagrangian advection. In *Discrete Modelling and Discrete Algorithms in Continuum Mechanics 2001*, Sonar T, Thomas I (eds). Logos Verlag: Berlin, 2001; 110–119.
90. Billings SD, Beatson RK, Newsam GN. Interpolation of geophysical data using continuous global surfaces. *Geophysics* 2002; **67**(6):1810–1822.
91. Chatelain P, Leonard A. Face-centred cubic lattices and particle redistribution in vortex methods. *Journal of Turbulence* 2002; **3**:046.
92. Barba LA, Leonard A, Allen CB. Numerical investigations on the accuracy of the vortex method with and without remeshing. *AIAA #2003-3426*, *16th AIAA Conference on Computational Fluid Dynamics*, Orlando, FL, June 2003.
93. Barba LA. Vortex method for high-Reynolds number flows: increased accuracy with a fully mesh-less formulation. *Ph.D. Thesis*, California Institute of Technology, 2004.
94. Melander MV, McWilliams JC, Zabusky NJ. Axisymmetrization and vorticity-gradient intensification of an isolated two-dimensional vortex through filamentation. *Journal of Fluid Mechanics* 1987; **178**:137–159.

