Journal of Computational Physics 229 (2010) 1292–1310

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Global field interpolation for particle methods

L.A. Barba [a,*], Louis F. Rossi [b]

[a] *Department of Mechanical Engineering, Boston University, United States*
[b] *Department of Mathematical Sciences, University of Delaware, United States*

ARTICLE INFO

Article history:
Received 11 June 2009
Received in revised form 8 October 2009
Accepted 20 October 2009
Available online 28 October 2009

Keywords:
Particle methods
Meshfree methods
Vortex methods
Radial basis functions
Gaussian function
Adaptive methods
Interpolation
Approximation
Preconditioning
Iterative solution methods
Dense linear systems
Reverse heat equation
Gaussian deblurring

ABSTRACT

A common problem in particle simulations is effective field interpolation. Field interpolation is any method for creating an accurate representation of a given continuous field by a linear combination of overlapping basis functions. This paper presents two techniques for field interpolation, based on a radial basis function (RBF) formulation using Gaussians. The application in mind is vortex methods, where one needs to determine the circulation (or strength) of individual vortex particles with known position and scale to represent a given vorticity field. This process is required both to initially discretize a given vorticity field, and to replace a configuration of particles with another for the purposes of maintaining spatial accuracy. The first technique presented is formulated as an RBF collocation problem, and obtains a solution accurately and with excellent algorithmic efficiency by means of a preconditioned iterative method. The preconditioner is a sparse approximation, based on localization, to the dense coefficient matrix of the RBF system. The second technique uses approximate solutions to the reverse heat equation, recognizing that the standard regularization used in vortex methods (estimating particle strengths using the local value of vorticity multiplied by particle area/volume) corresponds to a Gaussian blurring of the original field. A single time step is used, thus avoiding amplification of high frequencies, and accurate solutions are produced using explicit finite difference methods. Computational experiments were performed in two dimensions, demonstrating the accuracy and convergence of the proposed techniques. Application in three dimensions is straightforward, as radial basis function interpolation is neutral to dimension, but will require more computational effort.

1. Introduction

Particle and meshfree (or meshless) methods have received great interest in recent years and proven to be very successful for solving a variety of problems. They have a natural application in problems which are inherently discrete, such as the simulation of astrophysical systems or the dynamics of molecules or atoms out of equilibrium. In these applications there is no need for discretization, and the computational particles correspond to the physical objects under study. One method which originated in the astrophysical field and has found great success is the smoothed particle hydrodynamics (SPH) method. This method first appears in the literature in the late 1970s, and was reviewed in [23]. The SPH method uses particles as a discretization technique for a continuum formulation of the problem, and it has been extended to many other applications be-

* Corresponding author. Address: Boston University, 110 Cummington St, Boston, MA 02215, United States.
E-mail addresses: labarba@bu.edu (L.A. Barba), rossi@math.udel.edu (L.F. Rossi).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.031

yond astrophysics. It joins a variety of other meshfree methods where the particle representation is used as a means of numerical discretization. Another salient member of this group is the vortex particle method, where a particle representation of the vorticity field is formulated in a Lagrangian method used to solve the Navier–Stokes equations. The vortex method traces its origins earlier than SPH, with the regularized vortex particle approach being introduced in [12] but based on the singular vortex particle method first appearing in the classic work of Rosenhead [28]. An overview of the main computational and theoretical issues with the vortex method can be found in the book [13], while standard review papers of the early development of the method are [19] and [20].

The term "particle method" does not refer to a single algorithm but rather a diverse collection of computational techniques for solving evolution equations. Researchers analyze and develop different techniques for resolving issues related to the solution of partial differential equations with particles, including capturing diffusion, satisfying boundary conditions, capturing vortex stretching in three-dimensional flows and effective time-marching schemes for the particles. In this paper, we examine the problem of representing a field using particles and/or replacing one set of particles with another for the purpose of maintaining accuracy and stability in the overall simulation. This procedure corresponds to the general problem of finding a set of basis functions with their weights, such that their linear combination is able to reconstruct the field variable accurately. We will refer to this procedure as particle field interpolation, or simply *field interpolation*.

When a particle method is used as a numerical discretization of a continuum problem, one represents the field variable as a linear combination of localized basis functions (the particles). Overlap of the particles is necessary to accurately reconstruct a continuous field variable, a fact which has been amply recognized in the vortex method literature where convergence results rely on the overlap condition [7]. In practice, it has been observed that the spatial accuracy in the numerical representation of the continuous vorticity field using the vortex particles increases exponentially with decreasing overlap ratio, defined as the distance between particles divided by the characteristic size [5]. In a Lagrangian approach (moving basis functions), the initial particle configuration can become severely strained or sheared, leading to a deterioration of spatial accuracy in the continuum representation. Thus, most if not all investigators face the need of remediating configurations of particles so that the basis functions continue to overlap in space throughout the simulation.

While various investigators have developed techniques for replacing a deformed configuration of particles with one that is uniform or nearly so, this aspect of particle method simulation is seldom addressed in the literature. Possibly, spatial and temporal errors have dominated particle calculations and errors induced by the particle redistribution has been of lesser concern. Currently, vortex methods involving elliptical, deforming basis functions are available, providing fourth-order accuracy in space [31]. The accurate replacement of one particle configuration with another acquires greater relevance when maintaining accuracy in combination with such a method.

A central issue in particle field interpolation is that the associated linear systems of equations are ill-conditioned. In one of the first attempts to avoid inverting the ill-conditioned system, Beale proposed applying a residual correction method to interpolation problems with fourth-order kernels, which relies on having square lattices of particles [6]. Computational experiments using Beale's approach were reported in [11], here using Gaussian kernels on a fixed mesh with changing values of the core size. The importance of preconditioning in field interpolation was later recognized in [21]. However, tensor-product interpolation is a popular alternative that avoids solving the linear system altogether. In this method, a disordered set of particles is replaced with a set of uniformly distributed ones by projecting particle weights (circulation) onto a uniform mesh. The procedure is called remeshing or particle redistribution or re-initialization by different authors. One of its first successful uses was in the simulation of cylinder flow in [18], and the approach has since been at the heart of many important results with vortex particle methods, such as a recent calculation of aircraft wakes with *billions* of particles [10]. Alternative spatial adaptation techniques that have been introduced include the splitting and merging of particles [29,30]. However, the use of tensor-product interpolations continues to be the state-of-the-art in vortex methods. Some evidence is available suggesting that remeshing could pose an accuracy limitation on the overall calculations (experiments in [4] show a jump in the errors upon the first remeshing process), and for that reason we are interested in the alternatives. One alternative that has been recently proposed and carefully studied [5] is to re-project the field variable, which in the case of vortex methods is the vorticity, using the particles as interpolation centers. The technique is based on a radial basis function (RBF) formulation of the problem, but the issue of ill-conditioning was not investigated.

The motivation of the present work is providing a spatial adaptation technique appropriate for use with fourth-order viscous vortex methods with elliptically deforming bases [31]. However, the results are general and applicable to any interpolation problem utilizing Gaussian radial basis functions. The paper begins with an overview of the vortex method, and an explanation of how the present work advances us toward our goal. In Section 3, we overview some relevant aspects of the theory of radial basis function interpolation and issues related to the solution of the ill-posed reverse heat equation. In Section 4, we present methods of solution and numerical experiments for the field interpolation problem applied to the vortex particle method. We use mainly two experimental setups: one is a passive scalar streak which has been deformed by a differentially rotating flow, and evolved using the fourth-order elliptical vortex method [32]; the second is the high-gradient but smooth vortex patch of [22]. These two experimental setups are used to demonstrate several approaches to field interpolation, including: solving the dense linear system associated with an RBF interpolation problem using a preconditioned iterative method; and, using the reverse heat equation to deconvolve an initial guess obtained using local vorticity and particle volumes. We end with conclusions and indications of the questions left for future work.

2. Vortex methods and representing fields with smooth particles

The vortex method solves the Navier–Stokes equation at constant density in vorticity formulation, namely, the vorticity transport equation:

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

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 $\frac{D\omega}{Dt} = 0$, where $\frac{D}{Dt}$ stands for the material derivative. One can immediately see how a Lagrangian approach is ideal for such a problem: to satisfy this equation it suffices to allow small computational elements of vorticity to follow material trajectories. Three-dimensional vortex methods have been devised by formulating a particle solution to the vortex stretching/tilting term, while viscosity has been accounted for in a variety of methods. See [13] for a description of the standard methods, and [5] for an overview of viscous schemes including the core spreading method that we use.

The discretization of the vorticity is accomplished by representing the continuous field by a summation of particle-type elements, each consisting of a local distribution of vorticity ζ_σ with a particular strength, γ_i. Thus, the vorticity field is represented by a linear combination of basis functions, as follows:

$$\boldsymbol{\omega}(\mathbf{x}, t) \approx \boldsymbol{\omega}_\sigma(\mathbf{x}, t) = \sum_{i=1}^{N} \gamma_i(t) \zeta_\sigma(\mathbf{x} - \mathbf{x}_i(t)). \tag{2}$$

The particles have a characteristic spatial scale, σ, and their strength (representing circulation) is vector-valued in 3D and a scalar in 2D. Many choices have been proposed and used for the local distribution of vorticity, but one of the most common is the Gaussian:

$$\zeta_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-|\mathbf{x}|^2}{2\sigma^2}\right). \tag{3}$$

One of the important parameters in the vortex method is the so-called *overlap ratio*, representing the average distance between the particles relative to their characteristic scale; we can write the overlap ratio as $\beta = h/\sigma$. Proofs of convergence of the vortex method rely on the assumption $\beta < 1$.

Numerical experiments presented in [3] demonstrated how the error of the vortex particle representation decreases super-exponentially with the overlap value. Examples using Lamb–Oseen vortex distributions demonstrated an error in the vorticity field of order 10^{-7} for $\beta = 1$ and decreasing further down to machine error at $\beta = 0.7$. The plot in Fig. 1 shows the errors obtained when representing the Lamb–Oseen vorticity field with particles at a fixed particle separation, h, but varying particle width, σ, resulting in a varying overlap ratio. It can be seen how the errors decrease by several orders of magnitude as the overlap ratio decreases passing through the value $\beta = 1$. Thus, maintaining sufficient overlap during the course of a vortex computation when relative positions are shifting is essential for maintaining accuracy.

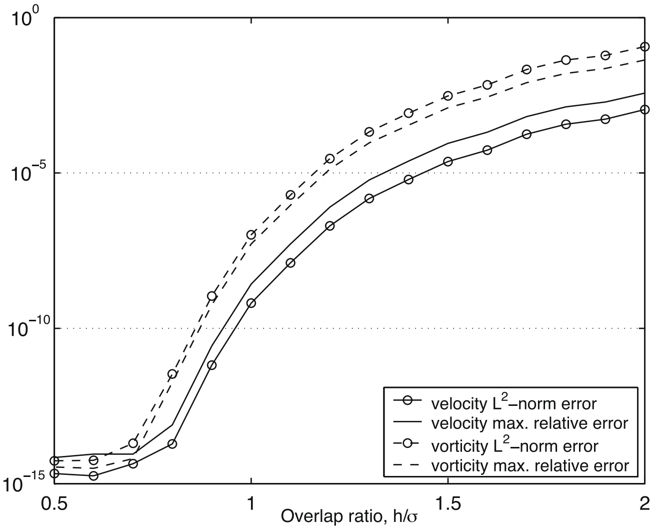


Fig. 1. Super-exponential dependence of the accuracy of representation with Gaussian particles, with respect to the particle overlap, $\beta = h/\sigma$. The test field corresponds to a Lamb–Oseen vortex, where the separation h was kept constant, but σ was varied each time (thus, the number of particles N is constant).

One of the remaining issues for ensuring numerical accuracy with the vortex method is the choice of the strengths, γ_i, which ensures this accuracy for a given initial condition. Historically, the local distribution function for the particles was introduced to regularize the formula for velocity as a function of vorticity, or Biot–Savart formula, given by

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

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

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

Introducing the discretized vorticity ω_σ for ω in the above formula, one obtains a summation over all particles to calculate the velocity at any evaluation point. In the point vortex method, the local distribution function is a delta function, and thus the velocity is singular. Perhaps surprisingly, the point vortex method will converge to the Euler equations, even without smoothing [16]. Nonetheless, the regularization with ζ_σ addresses the stiffness problem by smoothing the velocity kernel.

There are two interpretations of the computational representation used by vortex methods. The earlier interpretation is that the vorticity field is represented as a linear combination of delta functions, and the smoothed kernel allows one to compute an approximate, regular velocity field and avoid stiffness. The modern interpretation that we embrace is that the vorticity field should be represented as a linear combination of smooth basis functions, and that the smoothed kernel is an exact representation of the induced velocity field. The modern interpretation facilitates a more direct convergence formulation and the potential for new viscous vortex methods [31], but it restricts the admissible computational vorticity fields to those which can be represented as a combination of smooth functions.

The historic origin of the smoothing of particle distributions (resulting in the so-called vortex blob method) meant that vortex method practitioners have seldom looked at the problem of representing the vorticity field with, say, Gaussians, as an interpolation problem. In fact, the vortex method discretization is equivalent to a radial basis function interpolation problem. In [5] this fact was exploited to develop a method for spatial adaptation of the particles in a time-marching algorithm using radial basis function (RBF) interpolation.

The standard approach for initialization of a vortex method calculation has been to identify the smooth particle with an area (in 2D), and approximate this area to write $\gamma_i = \omega_i A_i$. That is, vorticity at the particle center multiplied by the particle area (or volume in 3D), gives its weight in units of circulation. When initializing particles on a square lattice, thus, the "standard initialization" in 2D gives the following simple estimation:

$$\gamma_i = \omega_i h^2 = \omega(\mathbf{x}_i)h^2. \tag{6}$$

This method of initializing the vortex particles incurs in an error called "smoothing error" or "regularization error." It was recognized early on that one could improve on the choice of γ_i's given by (6), for example using an iterative method as in the circulation processing approach of [6]. We will demonstrate how this error can easily dominate the accuracy of a vortex method calculation, which often results in workers choosing a much smaller core size σ than would really be needed, generating much larger problem sizes (large N) than necessary.

In fact, finding the particle strengths that best represent the vorticity at the particle locations amounts to solving the following interpolation problem:

$$\omega_\sigma(\mathbf{x}_j) = \omega(\mathbf{x}_j), \tag{7}$$

$$\omega_\sigma(\mathbf{x}_j) = \sum_{i=1}^{N} \gamma_i \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i), \tag{8}$$

Eq. (7) corresponds to the interpolation conditions, which are satisfied by (8) when solving the linear system of equations on the particle strengths,

$$\mathcal{A}\vec{\gamma} = \vec{\omega} \tag{9}$$

with,

$$\mathcal{A}_{ij} = \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i) \tag{10}$$

The possibility of a more accurate initialization solving the system (9) was recognized in [17], where successive over-relaxation (SOR) with under-relaxation was used to initialize calculations of elliptical vortex patches. Despite the successful calculations of [17], the problem of inverting the system (9) is far from being solved. The authors of [13] point out that it is very much a current research topic (p. 211). The difficulty in solving (9) arises from the fact that the matrix $\mathcal{A}$ is full and badly conditioned, and the size of the problem is very large for any applications of interest (production vortex codes regularly deal with 10^5 or 10^6 particles).

In [5], the RBF interpolation approach was used to re-discretize the vorticity field at intermediate times in a vortex method calculation, with the goal of preserving overlap despite the Lagrangian deformation of the particles. The method was first programmed in MATLAB and later ported to a parallel version using the PETSc library [2], and in both cases the method of solution used was the generalized minimal residual method (GMRES). High-accuracy results were demonstrated, but the behavior of the ill-conditioned system was not investigated.

Recently a new vortex method has been introduced which utilizes elliptically deforming Gaussian particles and thus achieves fourth-order accuracy in space [31,32]. This method offers the potential of being able to compute highly-deforming vortex flows with manageable numbers of particles, as the elements naturally adapt to the flow deformations. Truly adaptive calculations with the elliptical vortex method for indefinite times are possible if a spatial adaptation module is added. One needs to re-initialize the vorticity field at intermediate times in a simulation to control the aspect ratio of the computational elements [27]. The elliptical particles, however, preclude the use of standard remeshing schemes using tensor-product formulations, as these rely on an assumption of constant particle volume.

3. Solving global field interpolation for vortex particle representations

In this section, we will develop two techniques for particle field interpolation: preconditioned radial basis function collocation, and a deconvolution formulated with the reverse heat equation. We will compare these techniques with the common low-order practice that initializes the particle strengths as $\gamma_i = \omega_i h^2$. One common technique that has not been assessed in this paper is the use of tensor-product formulations with splines-based kernels. In the vortex method community, most workers use an interpolation kernel introduced for SPH (smoothed particle hydrodynamics) methods in [24], commonly known as M_4' interpolation. We do not make direct comparisons with M_4' interpolants because these methods are not suitable for elliptically deforming basis functions (they assume uniform particle geometry, i.e., axisymmetric bases with uniform radius), nor are they suitable for the projection of a known field onto a set of basis functions (as required in the initialization of a simulation).

3.1. Field interpolation with preconditioned RBF collocation

The particle discretization of the vorticity in the vortex method corresponds to a radial basis function (RBF) expansion of the field. Therefore, the initial solution for the particle strengths is equivalent to an RBF interpolation collocation problem, as illustrated by Eqs. (7) and (8). RBF interpolation was introduced as a tool for solving multivariate scattered data interpolation problems. There has been a great production of research results in this field, with some excellent books being published in recent years [8,14].

In this paper, we present a study of field interpolation using RBFs in regards to its accuracy and the behaviour of solution methods. We start by demonstrating a preconditioner, appropriate for rapidly decaying basis functions such as the Gaussian, based on a sparse approximation to the full coefficient matrix by localization. The preconditioned iterative solution is effective at producing convergence to a low tolerance in only a handful of iterations. Its motivation and definition are presented in this section, while numerical experiments are presented in Section 4.3.

As mentioned, the RBF interpolation problem results in an ill-conditioned linear system with a dense matrix. As such, the success of an iterative solution method relies entirely on the application of a preconditioner. Although the preconditioning of sparse linear systems has received significant attention [1,33], this is not the case for dense systems. For completeness, let us recall the basics of preconditioning. The problem at hand is the solution of a linear system

$$\mathcal{A}\vec{x} = \vec{b}, \tag{11}$$

where $\mathcal{A} = [a_{ij}]$ is an $N \times N$ matrix of coefficients and $\vec{b}$ is a right-hand-side vector. For very large values of N, the only feasible methods of solution are iterative, and the convergence of iterative methods depends critically on the conditioning of $\mathcal{A}$. Preconditioning applies an operator to Eq. (11) to transform the system into another one where the coefficient matrix has improved spectral properties. Applying an implicit left preconditioner $\mathcal{M}$ to the system (11) results in:

$$\mathcal{M}^{-1}\mathcal{A}\vec{x} = \mathcal{M}^{-1}\vec{b}. \tag{12}$$

The new system (12) of course has the same solution as the original one (11), but it should be easier to solve. One can also precondition from the right, or perform split preconditioning by transforming the original system from both sides.

An iterative method of solution will converge quickly if the preconditioned system matrix has eigenvalues that are clustered close together. Of course, if $\mathcal{M} = \mathcal{A}$, then the eigenvalues of the preconditioned system are all equal to 1, but this is of no use because transforming the system for preconditioning is as hard as solving it. So, one would expect that a good preconditioner $\mathcal{M}$ is in some way an approximation to $\mathcal{A}$.

In our problem of performing field interpolation of vorticity using particles, the system matrix is:

$$\mathcal{A}_{ij} = \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i) \tag{13}$$

with

$$\zeta_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\exp\left(\frac{-|\mathbf{x}|^2}{2\sigma^2}\right)$$

Because of the rapid decay of the Gaussian function, a sparse approximation to $\mathcal{A}$ can be constructed by making zero those elements that correspond to distances $|\mathbf{x}_j - \mathbf{x}_i|$ larger than a chosen threshold. Therefore, we propose as (implicit) preconditioner the following sparse matrix:

$$\mathcal{M}_{ij} = \begin{cases} \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i) & \text{if } |\mathbf{x}_j - \mathbf{x}_i| < R, \\ 0 & \text{if } |\mathbf{x}_j - \mathbf{x}_i| > R. \end{cases} \tag{14}$$

The goal is to determine the values of R for which such a preconditioner will be effective in providing fast convergence of an iterative solution method. If R is small, the sparsity of the preconditioner will be large (few non-zero values). However, the preconditioner may be ineffective. If R is large, the preconditioner will be less sparse, but it should have a greater chance of accelerating convergence. Of course, if R is as large as the domain size, the preconditioner is dense and equal to the coefficient matrix, thus its inversion is as hard as solving the initial system. It is not possible to guarantee *a priori* that a useful R value exists that provides sufficient sparsity and effective convergence acceleration. Thus, we investigate this problem experimentally. In Section 4.3 we will present numerical experiments utilizing RBF interpolation by collocation, with the linear system being solved by GMRES both without preconditioning and using the preconditioner presented above with various choices of R.

3.2. Field interpolation with the reverse heat equation

Another approach to determining the unknown circulation values, γ_i, of a set of particles describing a vorticity field is to improve upon the simple estimation (6). We note that this simple estimation maps ω to a distribution of γ's corresponding to a blurring of the exact vorticity field. If we let ω_σ refer to the blurred field induced by (6), we see that

$$\omega_\sigma = \sum_{i=1}^{M} \frac{\gamma_i}{2\pi\sigma^2} \exp\left(-\frac{|\vec{x}-\vec{x}_i|^2}{2\sigma^2}\right) \approx \int\int \frac{\omega(\vec{y})}{2\pi\sigma^2} \exp\left(-\frac{|\vec{x}-\vec{y}|^2}{2\sigma^2}\right) dA. \tag{15}$$

The approximation arises because we can interpret the RBF interpolant as a Riemann sum for the convolution (15). It is well known and easy to show that the blurred field on the right is an $\mathcal{O}(h^2)$ approximation to ω. The convolution (15) corresponds to the solution, $\omega(\vec{x}, t)$, to the unbounded heat equation

$$\partial_t \omega = \nabla^2 \omega, \quad \omega(\vec{x}, \sigma^2/2) = \omega_\sigma(\vec{x}), \tag{16}$$

at time $T = \sigma^2/2$. In other words, we associate the field ω with a blurred circulation. The circulation itself is not known, thus, we apply the reverse heat equation to recover it as the initial data leading to this solution. There are two mathematical and computational issues raised in proposing this technique. First, we need to understand the error in the approximation (15) and ensure that reversing the heat equation is an appropriate means of improving the accuracy of (6). Once we establish that (15) is a good estimate, we address the second relevant issue of efficiently reversing the heat equation to improve (6).

The approximation (15) is often taken for granted, but a precise examination is necessary if one is to understand the utility and limitations of a reverse heat equation method. Beginning with (15), we can subdivide the domain into rectilinear regions Δ_i over the support of ω. If ω does not have compact support, we assume it decays exponentially so that we can truncate ω with spectral accuracy,

$$\int\int \frac{\omega(\vec{y})}{2\pi\sigma^2} \exp\left(-\frac{|\vec{x}-\vec{y}|^2}{2\sigma^2}\right) dA = \sum_{i=1}^{M} \int\int_{\Delta_i} \frac{\omega(\vec{y})}{2\pi\sigma^2} \exp\left(-\frac{|\vec{x}-\vec{y}|^2}{2\sigma^2}\right) dA. \tag{17}$$

Defining

$$F(\vec{x}, \vec{y}) = \frac{\omega(\vec{y})}{2\pi\sigma^2} \exp\left(-\frac{|\vec{x}-\vec{y}|^2}{2\sigma^2}\right), \tag{18}$$

the integrands of (17) can be expanded about their centroids $\vec{y}_i$ as,

$$F(\vec{x}, \vec{y}) = F(\vec{x}, \vec{y}_i) + DF(\vec{x}, \vec{y}_i)[\vec{y} - \vec{y}_i] + \frac{1}{2} D^2F(\vec{x}, \vec{y}_i)[\vec{y} - \vec{y}_i, \vec{y} - \vec{y}_i] + \cdots, \tag{19}$$

where DF is the vector of partial derivatives of F with respect to the components of $\vec{y}$ and D^2F is the matrix of second partial derivatives of F. Each of these terms is a multilinear form as indicated, and under reasonably weak assumptions such as the continuity of ω, F will be analytic. Inserting (19) into (17), one can integrate the series term by term. For simplicity, we shall assume the Δ_i's are squares of width h, but it is easy to perform these calculations for rectangular or other tessellations;

$$\sum_{i=1}^{M} \int\int_{\Delta_i} F(\vec{x}, \vec{y}) d\vec{y} = \sum_{i=1}^{M} \left[F(\vec{x}, \vec{y}_i) h^2 + \frac{1}{24} h^4 \nabla^2_{\vec{y}} F(\vec{x}, \vec{y}_i) + \cdots \right]. \tag{20}$$

The DF term is odd and cancels out when integrated over a symmetric interval. The first term on the right in (20) corresponds to (6). The next term on the right is the leading-order source of the error in the approximation (15). This error term can be rewritten as an integral:

$$\varepsilon = \sum_{i=1}^{M}\left[\frac{1}{24}h^4\nabla_{\vec{y}}^2 F(\vec{x},\vec{y}_i)\right] \approx \frac{h^2}{24}\int\int \nabla_{\vec{y}}^2 F(\vec{x},\vec{y})dA = \frac{h^2}{24}\oint(\nabla_{\vec{y}}F(\vec{x},\vec{y}))\cdot\hat{n}dl \tag{21}$$

where the latter integral is a line integral around the boundary of the domain. We assume that ω is at least twice differentiable in the estimate above. The regularity of ω has a significant impact on the convergence rate. If ω and its derivatives vanish on the boundary, the entire contribution is zero, and the residual difference between the initial sum and approximate integral would dominate the expression. What is to stop us from continuing this process indefinitely and crossing out successive integrals in the Taylor expansion? Nothing except the regularity of ω. If ω is not analytic, then the Taylor series expansion is not valid throughout the domain, and there will be nontrivial contributions to the error from terms corresponding to jumps in the derivatives. Thus, we expect convergence of order (h^{p+2}) where p is the maximum number of continuous derivative of ω in the domain being interpolated. If ω is smooth as in the case of moderate Reynolds number flows, then ω is exponentially small but nonzero near the boundary. In this case, we see that (6) converges exponentially to a Gaussian blurring of the field ω. In the two model problems that we explore in this paper, the first (convection–diffusion) is analytic and the second (vortex patch) is not analytic.

Numerical methods for reversing the heat equation are known to be problematic and the equation is famously ill-posed. One way to understand the issue is to examine the action of the PDE in the Fourier domain. If

$$\omega = \int\int \hat{\omega}(\vec{k},t)e^{i\vec{k}\vec{x}}d\vec{k}, \tag{22}$$

then the Fourier transform satisfies

$$\frac{\partial\hat{\omega}}{\partial t} = -|\vec{k}|^2\hat{\omega}, \quad \hat{\omega}(\vec{k},\sigma^2/2) = \widehat{\omega_\sigma(\vec{x})}. \tag{23}$$

In other words, the growth rate of high-frequency modes increases to infinity as the frequency approaches infinity. This poses a significant challenge if $\sigma^2|\vec{k}|^2$ is large. However, we shall see that for small values, one can numerically integrate the reverse heat equation with high accuracy.

The reverse heat equation and shock filtering has a prominent and successful history in deblurring algorithms [26,15]. The dominant technique when reversing the heat equation is to impose an a priori bound on the solution or its variation as a penalty function [9,34,25]. Another approach is to introduce numerical diffusion as an additional low-pass filter to regularize direct solutions (see [15] for example). Unlike image enhancement applications, our goal is not to enhance high frequency details or other aspects that make image interpretation easier to perform. The goal of field interpolation is to accurately compute ω from ω_σ. In the world of image deblurring, this would correspond to producing a low signal-to-noise (SNR) ratio. For instance, in [15] an algorithm is presented with an SNR of 18.4 which would correspond to a relative error of roughly 1.45% for a blurred test image.

For the purposes of field interpolation, we assume that the vorticity can be resolved with overlapping Gaussian basis functions of core size σ so that σ will be smaller than the length-scale of variations in the vorticity field. When this is the case, it is not necessary to integrate (16) for long time intervals. In fact, we circumvent the stability issues associated with numerically integrating an ill-posed PDE by taking one step in time. This is justified because the limiting computational parameter is the core size σ. For the reverse heat equation process, the total duration of the integration is $T = \sigma^2/2$, so that spatial resolution is linked to the duration of the computation. Even though we take only one step in time, the field interpolation is consistent because the duration of the temporal integration collapses ($T \to 0$) as we refine the spatial resolution ($\sigma \to 0$).

There are many ways to solve simple linear PDEs such as the reverse heat equation. In Section 4.3, numerical experiments are presented where the reverse heat equation is solved using finite differences. Using an mth order discretization in time and an nth order discretization in space, we expect an accuracy of $\mathcal{O}(k^m) + \mathcal{O}(h^n) = \mathcal{O}(\sigma^{2m}) + \mathcal{O}(\sigma^n) = \mathcal{O}(\sigma^{\min(2m,n)})$ when calculating the circulation, where k is the time interval and h is the mesh spacing. Therefore, we expect the error in the induced vorticity field to be $\mathcal{O}(\sigma^{\min(2(m+1),n+2)})$. For the Laplacian, we will use second-, fourth- and sixth-order symmetric finite differences. Of course, boundary conditions are required. For simplicity of implementation, we use Dirichlet boundary conditions throughout this paper. Some of the limitations of this choice are discussed in Section 4.4, but we leave a thorough examination of boundary conditions for future work. For the temporal integration, we will present a low-order demonstration using Euler's method and a high-order demonstration using fourth-order Runge–Kutta. In the low-order demonstration, the procedure is as follows.

1. Select new basis function positions on a regular mesh $\vec{x}_{i,j}$. Here we deviate slightly from our convention of indexing basis functions sequentially. We represent the new basis function positions with a double index (i,j) which also represents their location on a discrete mesh $x_i = x_0 + ih, y_j = y_0 + jh$ where h is the mesh spacing and (x_0, y_0) is the lower left corner of the grid.
2. Calculate $\omega(x_i, y_j)$ at the new basis function positions and then calculate the initial circulations $\gamma_{i,j}^{(0)}$ using (6).
3. Take one step in time to integrate the reverse heat equation.

$$\gamma_{i,j}^{(1)} = \gamma_{i,j}^{(0)} + \frac{\sigma^2}{2h^2}\left(\gamma_{i+1,j}^{(0)} + \gamma_{i-1,j}^{(0)} + \gamma_{i,j-1}^{(0)} + \gamma_{i,j+1}^{(0)} - 4\gamma_{i,j}^{(0)}\right) = \gamma_{i,j}^{(0)} + \frac{1}{2\beta^2}\left(\gamma_{i+1,j}^{(0)} + \gamma_{i-1,j}^{(0)} + \gamma_{i,j-1}^{(0)} + \gamma_{i,j+1}^{(0)} - 4\gamma_{i,j}^{(0)}\right). \tag{24}$$

For this method, $m = 1$ because we are taking only one step of Euler's method so we use the local truncation error and $n = 2$. Therefore, we expect the scheme to be $\mathcal{O}(\sigma^4)$. For boundary constraints, one is free to implement Dirichlet or Neumann conditions. Any of the finite-difference implementations noted above is useful in particle simulations because they are fast, explicit and require very little memory. Thus, one can easily replace one configuration of basis functions with another without expending a great deal of time or allocating large amounts of memory. We also note that while we are using a regular mesh, we are not required to use a rectangular domain. For instance, for initial values of $\gamma^{(0)}$ that are very small at some locations, one is free to exclude them and treat them as zero in (24) so that new particles conform to the support of the field ω. In Section 4.4, we will explore the performance of this technique using a variety of differentiation stencils and time-stepping schemes.

4. Numerical experiments of field interpolation

4.1. Two model problems

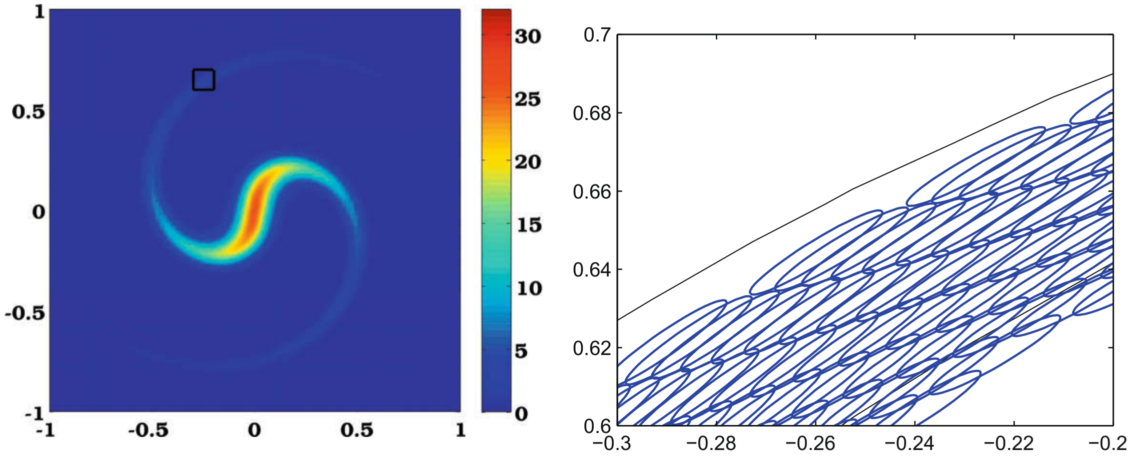
To study the efficacy of the two field interpolation methods suggested, we will use two test fields. The first test field consists of data obtained from a numerical calculation of a convection–diffusion problem, using the elliptical vortex method [32]. Thus, the field is the superposition of deformed elliptical Gaussian basis functions, with aspect ratio a. The elliptically deforming vortex particles have been evolved for a number of time steps, at the end of which the greatest aspect ratio is 6.2. Fig. 2(b) shows the location of the elliptical particles on an area of detail in the original field; as shown, there is adequate overlap of particles to resolve this computation. The deformation of the basis functions helps maintain this overlap beyond what would be possible with axisymmetric computations, but at the cost of having highly anisotropic particles (see [32] for more details of this calculation). In this case, the field is not available as an analytic expression. The example represents the situation in which a practitioner might reach this point in a calculation and need to interpolate the induced field onto a new arrangement of basis functions to maintain spatial accuracy.

The second test problem is a circular vortex patch that smoothly transitions from zero to a maximum value; it was used in Ref. [22] to study vortex axisymmetrization (we will call this test field MMZ to abbreviate). Defining an inner radius R_i and an outer radius R_o, then, g_{axi} is a C^∞ function transitioning from 0 to C, defined as follows:

$$g_{\text{axi}}(r, R_i, R_o) = C\begin{cases} 1, & r \leqslant R_i, \\ 1 - f\left(\frac{r-R_i}{R_o-R_i}\right), & R_i < r < R_o, \\ 0, & r \geqslant R_o, \end{cases} \tag{25a}$$

$$f_\kappa(r) = \exp\left[-\frac{\kappa}{r}\exp\left(\frac{1}{r-1}\right)\right], \quad 0 \leqslant r \leqslant 1 \tag{25b}$$

$$\kappa = \frac{1}{2}e^2\ln(2) \tag{25c}$$



(a) Test function for field interpolation, obtained numerically.

(b) Basis function positions and shape inside the small square shown on 2a.

Fig. 2. First test function for field interpolation. The plot on (a) shows the entire field, obtained numerically with the elliptical vortex method after a certain time. The plot on (b) depicts the basis functions in the area of detail marked by a rectangle in (a). For the sake of clarity, the ellipses representing basis functions are drawn with a semi-major axis of $(1/5)\sigma a$ and a semi-minor axis of $(1/5)\sigma/a$ where σ is the core size and a^2 is the aspect ratio of the basis functions.

This test problem presents a smooth, uniform, axisymmetric gradient. We select $R_o = 0.95$ and $R_i = 0.6 \times 0.95$ for most numerical tests below (unless otherwise noted). The constant C is chosen so that the distribution has unit circulation, $\int\int g_{\text{axi}} = 1$. To express this function as a linear combination of basis functions, one needs to adequately resolve the gradient. With an analytic expression available, a practitioner interested in using this as initial data would need an effective way to accurately project it onto a collection of basis functions.

4.2. Numerical parameters

While both the RBF collocation and the reverse heat equation methods solve the same problem, the parameters that control their behavior and performance differ somewhat. Any field interpolation scheme requires σ, the basis shape parameter, as an input. Similarly, the overlap ratio $\beta = h/\sigma$ is determined by the requirements of the vortex method itself, since the overlap ratio has a direct impact on the spatial accuracy of the method over time. Different flow calculations and different numerical schemes may require different values of β, so this parameter too must be treated as an input to any field interpolation method. Unique to our preconditioned RBF collocation is R, the cutoff distance which dictates the sparsity of the preconditioner. Unique to the reverse heat equation technique are the temporal and spatial convergence qualities of the solver. In the following experiments, we explore these features and characteristics carefully and report on our findings of their performance.

Before demonstrating the solution methods that we propose and investigate in this paper, it is worthwhile to discuss the efficacy of the common method of initialization used in vortex method calculations. This consists of estimating the particle weights simply as the product of the local vorticity value and the particle area (2D) or volume (3D): $\gamma_i = \omega_i h^d$, where d is the dimension. The convergence rate of this initialization method is $\mathcal{O}(h^2)$ or $\mathcal{O}(\sigma^2)$ assuming β remains fixed. Also, it requires that the particles be located on a uniform, Cartesian mesh. Furthermore, the accuracy of this simple initialization method is inferior to that of either of the methods described in this paper. Numerical evidence is presented in the following sections.

4.3. Numerical experiments with preconditioned RBF interpolation

The motivation of this work is to progress toward a fully adaptive method by a combination of the fourth-order elliptical vortex method and a scheme for spatial adaptation of the particles using global field interpolation. As a starting point, we perform experiments consisting of a static re-projection of the vorticity field obtained after computing a problem with the elliptical vortex method. The test problem for the first experiment is a streak of vorticity that has been strained in a spiral due to a background differentially rotating flow, from [32]. We first obtain the vorticity values on a lattice covering 1.2 times the original bounding box of the elliptical particles. The locations with a vorticity value smaller than a given threshold are discarded, and axisymmetric vortex particles are laid onto the remaining locations, selecting a value of σ and h. The next step is to solve the linear system for the particle circulation strengths, such that the new set of axisymmetric particles reproduces the vorticity field of the old set of elliptical particles as accurately as desired.

4.3.1. Experiment #1

With parameters shown in Table 1, the initialization using $\gamma_i = \omega_i h^2$ results in a maximum point-wise error in vorticity of 12%, with respect to the maximum vorticity, $\omega_{\max} = 25.9$. This indicates that the value of σ chosen is rather large and considerable blurring is produced by the regularized particles. However, this is unimportant for a simple proof of concept experiment. To produce a more compact field of particles, locations are discarded where the original vorticity field has a value of less than the maximum value on the edge of the new bounding box. The resulting number of particles is $N = 1956$; see Table 1 for a summary of the experiment parameters and results. Table 1 also shows the results in terms of the maximum field interpolation error in the vorticity for the different calculations. The error measure is simply the point-wise difference between the correct vorticity value and the vorticity obtained by the superposition of the new particles, normalized by the maximum vorticity:

Table 1
Numerical parameters and results for Experiment #1. The field interpolation error, ϵ, refers to the maximum point-wise vorticity error, normalized by maximum vorticity.

Parameters	Maximum field interpolation error
$\sigma = 0.02828$,	$\gamma_i = \omega_i h^2 \rightarrow \epsilon = 0.12$
$h = \sigma$,	Unpreconditioned GMRES $\rightarrow \epsilon = 0.0004$
$N = 1956$	Preconditioned, $R = 3\sigma \rightarrow \epsilon = 0.0059$
GMRES max. iterations, 50	Preconditioned, $R = 3.75\sigma \rightarrow \epsilon = 2.06 \times 10^{-9}$
Restart iterations, 5	Preconditioned, $R = 4.5\sigma \rightarrow \epsilon = 2.17 \times 10^{-10}$
Exit tolerance, 10^{-10}	Preconditioned, $R = 6\sigma \rightarrow \epsilon = 1.18 \times 10^{-10}$

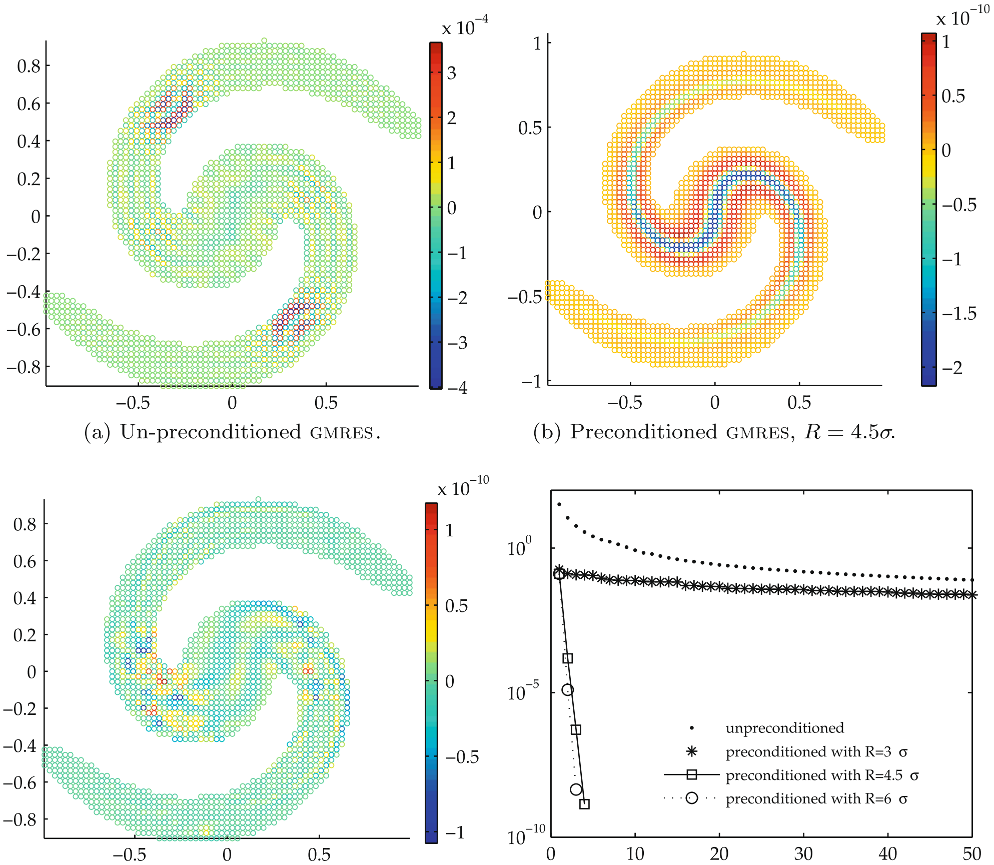
$$\text{error}: \quad \epsilon = \frac{\left|\omega_i - \sum_{j=1}^{N} \gamma_j \zeta_\sigma(x_j - x_i)\right|}{\omega_{\max}}. \tag{26}$$

The different calculations correspond to, first, an un-preconditioned GMRES solution, then, various preconditioned solutions with different values of R. The un-preconditioned solution exited without converging, and the resulting point-wise error at the particle locations is shown in Fig. 3(a). We note that the oscillations in the error seen in Fig. 3(a) point to the stability difficulties of the iterative method for this badly conditioned system. Nevertheless, the un-preconditioned GMRES solution results in a two-order of magnitude improvement in the maximum vorticity error, compared with the initial guess of $\gamma_i = \omega h^2$.

The error of the vorticity field at particle locations after the preconditioned solutions is shown in Fig. 3(b) and (c), for two choices of R which proved effective. Fig. 3(d) shows the relative residual norm within iterations; it can be seen that only two or three iterations were necessary for convergence (depending on the choice of R, as shown). The nonzero patterns of two preconditioning matrices are shown in Fig. 4. With $R = 3$, which was not an effective preconditioner, the sparsity is 1.2% non-zero values; with $R = 6$, there are 4.2% non-zero values. We looked at the conditioning of the resulting linear system matrices using the rcond() function of MATLAB. This function returns an estimate of the reciprocal of the condition number, so values close to zero indicate a badly-conditioned matrix, and values close to 1 indicate a well-conditioned matrix. The original coefficient matrix $\mathcal{A}$ returns a value of rcond equal to 2.13×10^{-4}, while the value of this estimate for $\mathcal{M}^{-1}\mathcal{A}$ is 4.76×10^{-4} with $R = 3\sigma$ (clearly no improvement) and 0.999 with $R = 6\sigma$.

4.3.2. Experiment #2

This test uses the MMZ vortex with the parameters given in Section 4.1 for the initial vorticity distribution. We place vortex particles on the nodes of a lattice of $2^n + 1$ divisions in each direction, covering the $[-1, 1] \times [-1, 1]$ domain. We use $n = 6$,



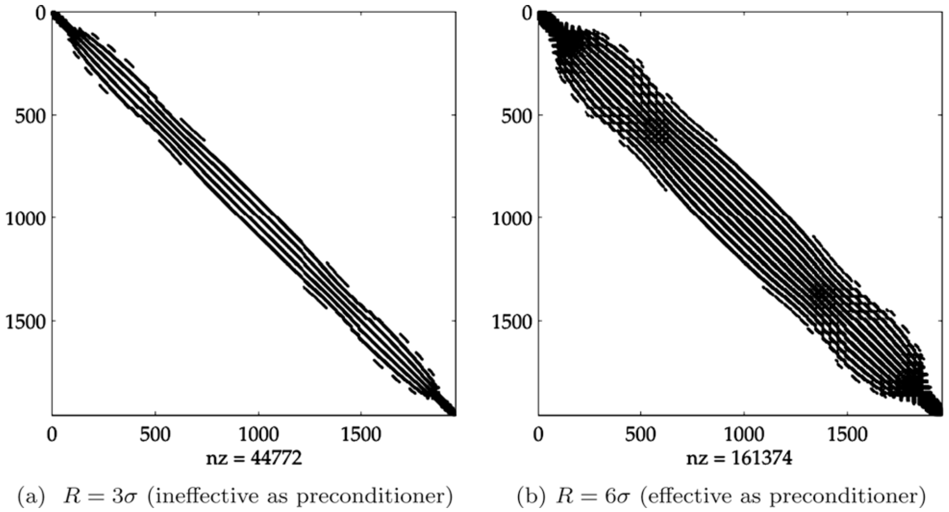
(a) Un-preconditioned GMRES.

(b) Preconditioned GMRES, $R = 4.5\sigma$.

(c) Preconditioned GMRES, $R = 6\sigma$.

(d) Relative residual norm in the GMRES iterations, for various cases.

Fig. 3. Results for Experiment #1. (a) Error in the total induced vorticity at the particle locations after solving for particle weights using un-preconditioned GMRES. (b) Error after solving the preconditioned GMRES system, as described in the text, using the sparse approximation preconditioner with $R = 4.5\sigma$, and (c) with $R = 6\sigma$. (d) Relative residual norm in the GMRES iterations, calculated in MATLAB as norm(b-A*x)/norm(b), for the different cases described in the text.



(a) $R = 3\sigma$ (ineffective as preconditioner) (b) $R = 6\sigma$ (effective as preconditioner)

Fig. 4. Nonzero pattern of the preconditioner matrices.

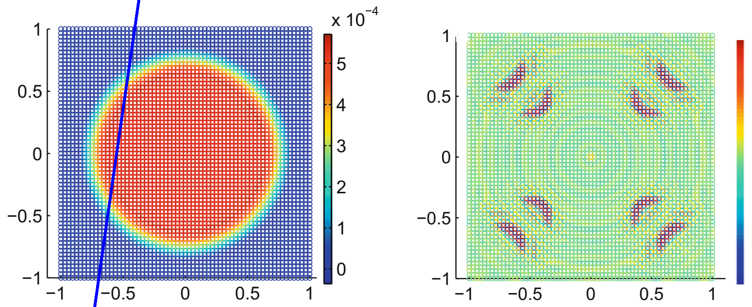
which results in $N = 4225$ particles with $\sigma = 0.03125$; see Table 2 for a summary of the experiment parameters and results. The maximum error in vorticity when using the initial γ estimate is $\mathcal{O}(10^{-2})$. Using the full coefficient matrix in a GMRES solution results in a reduction of the maximum vorticity error from $\mathcal{O}(10^{-2})$ to $\mathcal{O}(10^{-5})$; the resulting point-wise vorticity error is plotted in Fig. 5(b). In other words, 50 iterations of GMRES improve on the initial guess for γ, providing three orders of magnitude more accuracy in the discretized vorticity. However, it can be seen in Fig. 5(e) that GMRES is stagnating. Interestingly, using the sparse approximation of $\mathcal{A}$ as coefficient matrix in the GMRES calculation (instead of as a preconditioner) gives results which are of similar accuracy as the un-preconditioned GMRES with the full matrix. The values of the norm of the residual vector in the GMRES iterations lie practically on top of each other: the dots in Fig. 5(e) correspond to the sparse approximation used as coefficient matrix, while the open circles correspond to the un-preconditioned GMRES with the full matrix. Moreover, the difference in the solutions, i.e., the values of γ, is very small. The take-home message is: solving the sparse approximate linear system, obtained by neglecting the long-range effects of the Gaussian basis function, will provide an immediate improvement over the simple estimation $\gamma_i = \omega_i h^2$ which is used so often.

We now study the mesh effects observed on the scatter plot of vorticity errors in Fig. 5(b), corresponding to an un-preconditioned GMRES solution. To be precise, we want to confirm that the zones of larger error seen on Fig. 5(b) exhibiting four-fold symmetry are, in fact, mesh effects. To do this, we have performed a similar experiment, with the difference that the collocation points for the vortex particles lie on the nodes of a triangular lattice. For a similar initialization as in the square lattice, where the $[-1,1]^2$ domain is divided by $2^n + 1$ segments on each direction, we need to apply a correction to the lattice spacing $h = 1/2^{n-1}$ to produce a similar particle density. In the triangular lattice, we “stretch” the value of h to produce equivalent particle cell-area. For $n = 6$, this process results in $N = 4270$ on a triangular lattice; the equivalent square lattice had $N = 4225$. Initializing the γ values by means of an un-preconditioned GMRES solution results, in this case, in a vorticity error of the same order of magnitude as on the square lattice, but exhibiting six-fold symmetry rather than four-fold symmetry. See Fig. 6(a). Preconditioning with the sparse approximate matrix, with a threshold of $R = 6\sigma$, is similarly effective compared with the square lattice; see Fig. 6(b) and compare with Fig. 5(c).

Possibly, the mesh-influenced oscillations in the point-wise error seen in the un-preconditioned iterative RBF solutions are associated with a lack of resolution, or with a difficulty of the Gaussian basis functions with the given σ to resolve the gradients of the field. To test this hypothesis, we repeat the initialization of an MMZ function on a triangular lattice,

Table 2
Numerical parameters and results for Experiment #2. The field interpolation error, ϵ, refers to the maximum point-wise vorticity error, normalized by maximum vorticity.

Parameters	Maximum field interpolation error
$\sigma = 0.03125$,	$\gamma_i = \omega_i h^2 \rightarrow \epsilon = 0.027$
$h = \sigma$,	Unpreconditioned GMRES $\rightarrow \epsilon = 6.8 \times 10^{-5}$
$N = 4225$	Sparse matrix as system matrix with $R = 6\sigma \rightarrow \epsilon = 6.8 \times 10^{-5}$
GMRES max. iterations, 50	Preconditioned, $R = 3\sigma \rightarrow \epsilon = 1.3 \times 10^{-5}$
Restart iterations, 5	Preconditioned, $R = 4.5\sigma \rightarrow \epsilon = 1.7 \times 10^{-8}$
Exit tolerance, 10^{-7}	Preconditioned, $R = 6\sigma \rightarrow \epsilon = 7.4 \times 10^{-9}$

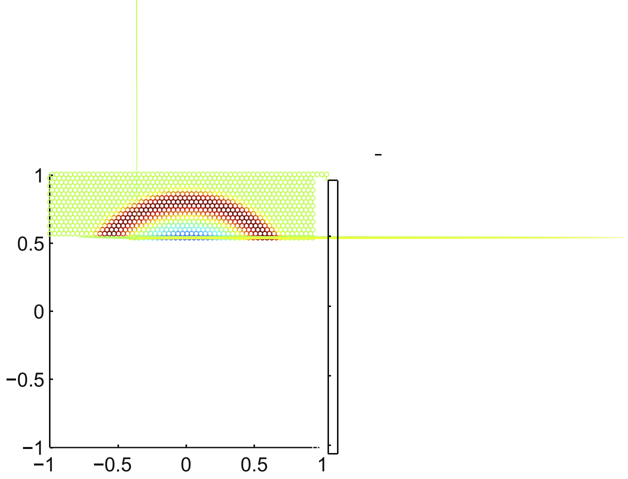


(a) Scatter plot of γ values.

but changing the slope of the function. The case we have used so far, described in Section 4.1, has $R_o = 0.95$ and $R_i = 0.6 \times 0.95$. The "relative steepness" of the MMZ function is defined by:

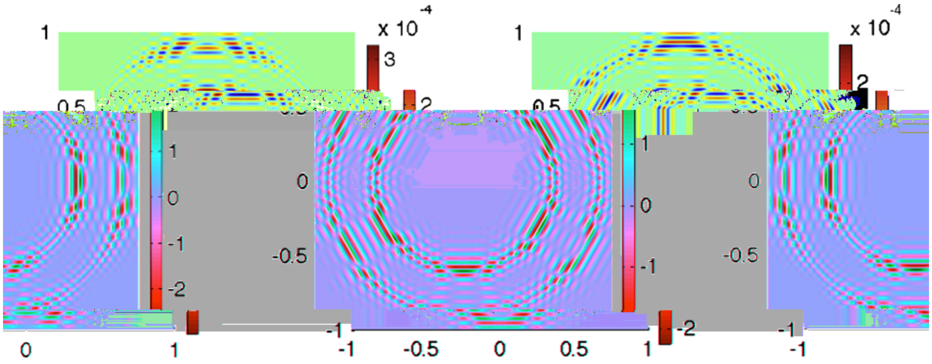
$$\delta = \frac{R_o - R_i}{R_o}. \tag{27}$$

Keeping $R_o = 0.95$, we consider three cases: Case A, $\delta = 0.4$ (presented previously); Case B, $\delta = 0.7$ (less steep); and Case C, $\delta = 0.95$ (least steep). Case B is a slightly less steep MMZ function compared to the case of the previous experiments. When



calculating the solution of the linear system for the γ values using un-preconditioned GMRES, we obtain areas of error reflecting the mesh-symmetry, as before, but the errors are one order of magnitude smaller than for Case A; see Fig. 7(b) and compare with Fig. 6(a). In fact, looking at Fig. 7(b), it appears that the areas of larger error are related not with the gradient of the function being interpolated, but with the curvature (second derivative).

Case C is a much less steep MMZ function, with the vorticity gradually falling from the maximum value at radii less than $R_i = 0.05$ to zero at $R_o = 0.95$ (see Fig. 8). In this case, again we observe a decrease of one order of magnitude in the vorticity error when using the un-preconditioned GMRES solution, with respect to Case B (with identical GMRES parameters). Similarly to the previous case, the areas of high error values appear to be located where there is high curvature of the field. What we have shown, then, is that the accuracy that can be obtained by a simple GMRES iteration without preconditioning does



ı a fine mesh for the precondi- 6σ, on a square lattice.

(b) Relative error on a fine mesh for the precondi- tioned solution, $R = 6\sigma$, on a triangular lattice.

(a) Relative error on tioned solution, $R =$

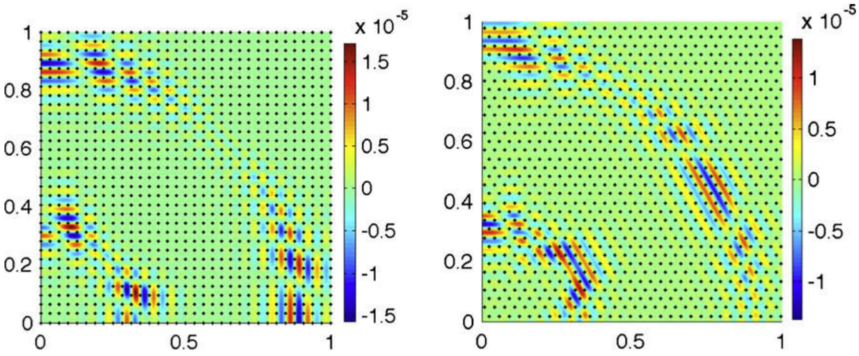
Fig. 9. Fine mesh errors of the discretization of the MZM function, Case A. Compare with Fig. 5(d).

depend on the features of the field. In this case, we have used the same width of the Gaussian basis functions, σ, and the same overlap ratio, resulting in the same number of particles in the fixed domain, but varied the steepness of the MMZ function each time. For less steep functions, the interpolation accuracy was improved over the more steep cases.

Finally, a potential stumbling block for the custom preconditioning of the RBF solution is found by plotting the vorticity errors on a very fine sampling mesh, thus revealing ‘‘sub-grid” effects. Consider Fig. 5, corresponding to the experiment with the MMZ function, Case A. The maximum vorticity errors obtained with the un-preconditioned solution are $\mathcal{O}(10^{-5})$, while the preconditioned solution gives $\mathcal{O}(10^{-9})$. Now, we calculate the vorticity field induced by the particles at a very fine sampling mesh, of size 1024^2, and obtain the error with respect to the analytical field. This is plotted in Fig. 9(a), while Fig. 9(b) presents the result of using a triangular lattice for the particle locations. As we can see, the maximum error at the sub-grid level is much larger than at the collocation points; here, it is $\mathcal{O}(10^{-4})$. Both the magnitude of the errors and the spatial structure look similar in the cases of un-preconditioned and preconditioned iterative solutions. Hence, the custom preconditioner ensures a very accurate interpolation (small field error at the collocation points), but does not guarantee an improvement at locations between particles.

Fig. 10 shows a zoom-in of the fine-mesh vorticity errors, for both the square and triangular lattices, when discretizing Case B of the MMZ function. The plots show the collocation points (particle locations) as black dots superposed to the color plot of the error. It can be seen clearly that the various peaks in the field error occur in the tracks between particles. In particular, it seems that the maximum errors appear where the gradient of the vorticity field aligns itself with the mesh axes of symmetry.

Although the larger errors at the mid-particle locations present a potential pitfall for particular interpolation applications, this problem may not have a severe effect in a Lagrangian particle method such as the vortex method. In this case, only the field values at the collocation points are used in the numerical calculations. In particular, in the vortex method the value of the vorticity at the particle locations is used to calculate the velocity, again at the particle locations. Given that the velocity is a convolution of the vorticity, the sub-grid errors may play a part. Probably, a combination of the limitations of using the Gaussian basis functions to represent a given vorticity field with the quadrature errors in the calculation of the velocity will finally have an effect in the accuracy of the calculated particle velocities. Thus, the field interpolation accuracies obtained



(a) Square lattice.

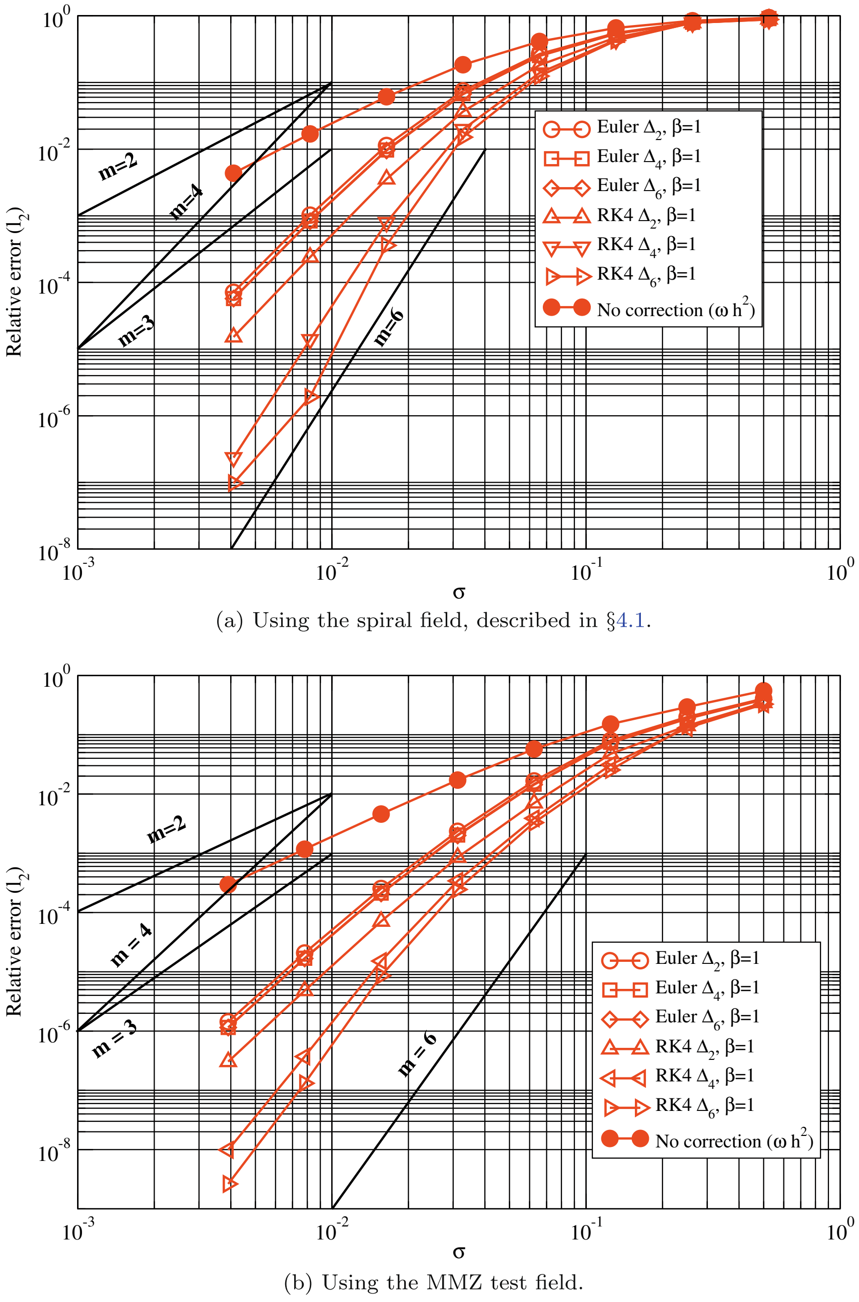
(b) Triangular lattice.

Fig. 10. Zoom-in of the fine mesh errors of the discretization of the MMZ function, Case B, with the collocation points.

with our preconditioned RBF method must be viewed with some caution. Nevertheless, our preconditioned RBF method offers extremely good algorithmic efficiency (number of iterations to converge).

4.4. Improving an initial guess via the reverse heat equation

In this subsection, we explore the reverse heat equation RHE technique for field interpolation by varying its two key ingredients, temporal and spatial discretization. We also look at the role of overlap ratio β knowing that β is chosen based on requirements not under the control of the field interpolation scheme but rather based on the desired accuracy of the particle method. The field interpolation technique must cope with β as chosen by the vortex method practitioner for a given flow simulation.



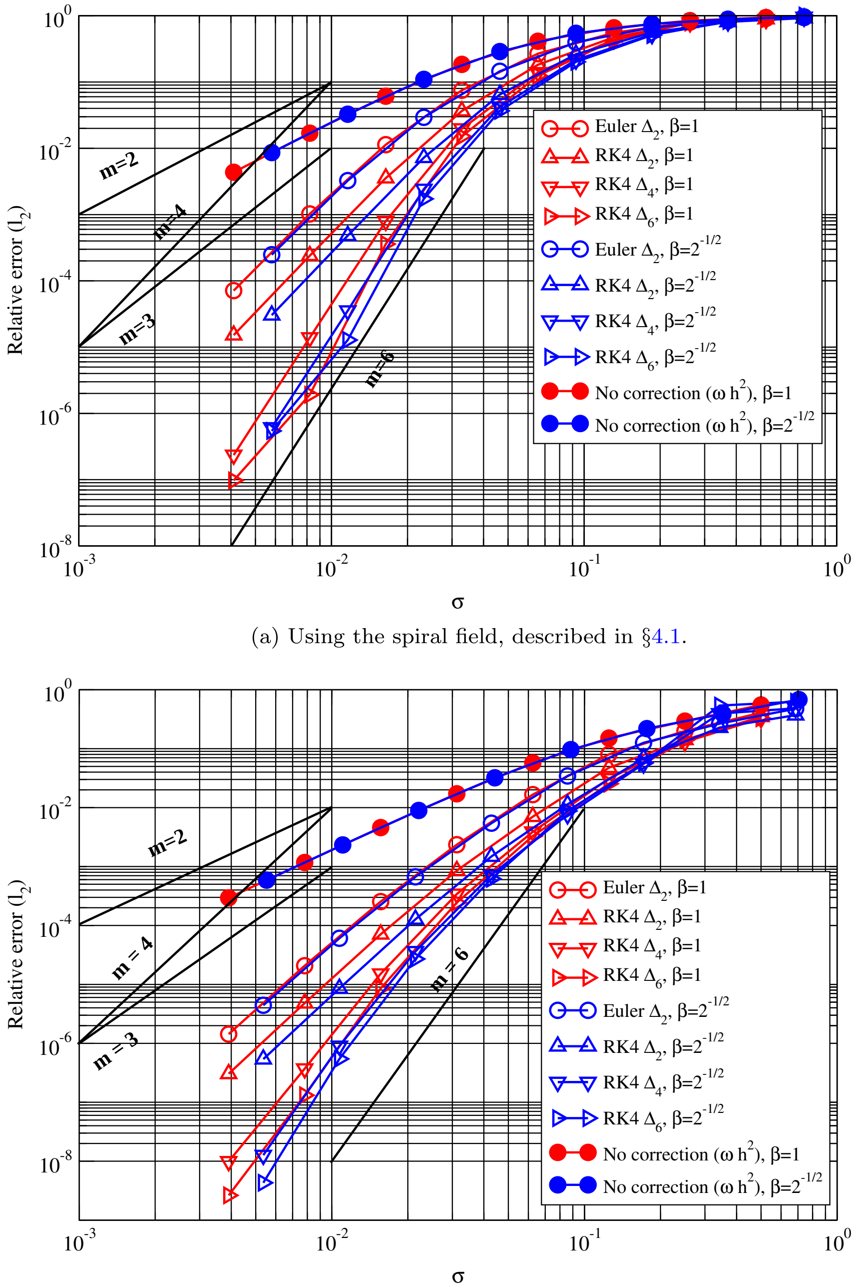
(a) Using the spiral field, described in §4.1.

(b) Using the MMZ test field.

Fig. 11. Convergence rates of reverse heat equation solutions with various differencing methods in space and time, using the two test fields described in Section 4.1.

To examine the efficacy of solving the reverse heat equation to improve on the initial guess for the γ values, we discretize our two base cases on a fine 1024^2 mesh as a reference solution. Then we apply field interpolation techniques for particles on submeshes of the base mesh. We always calculate the error on the finest mesh. Note that, unlike the RBF interpolation approach, the reverse heat equation solver is grid-based, because we implement standard PDE methods on a mesh. To understand the influence of temporal integration schemes, we implement a simple Euler's method integrator and a fourth-order Runge–Kutta integrator. For spatial discretization, we use second-, fourth- and sixth-order symmetric finite difference operators. In each case, we take one step in time to solve (16).

Fig. 11 presents a numerical convergence study of the different RHE schemes that were implemented. The first three runs show that the error is dominated by the Euler integrator; these are the first three points from the right of each curve. The overlap ratio β is kept fixed across all the data shown. From the earlier discussion, we recall that the convergence rate could

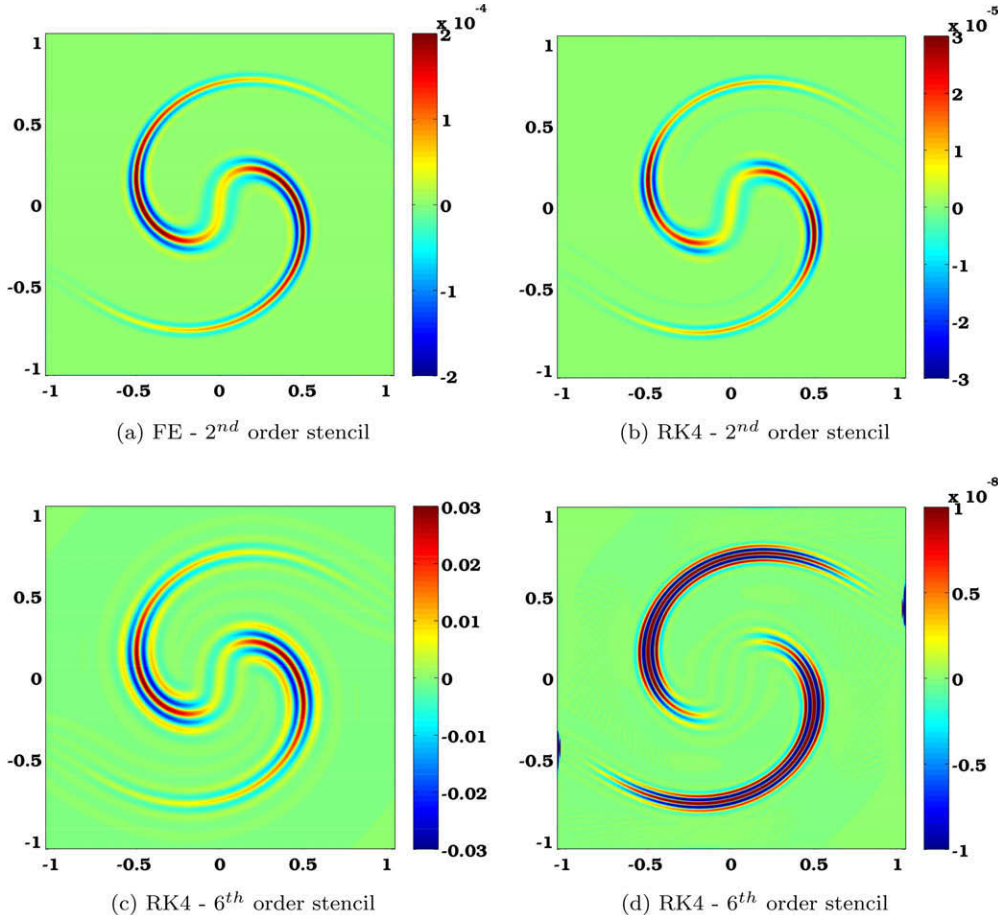


(a) Using the spiral field, described in §4.1.

(b) Using the MMZ test field.

Fig. 12. Convergence rates of reverse heat equation solutions and the impact of the core overlap ratio β for the two test problems.

be as great as $\mathcal{O}(\sigma^{\min(2(m+1),n+2)})$ where m is the order of the temporal integrator and n is the order of the spatial discretization. The spatial discretizations do not offer any help because Euler's method limits us to $\mathcal{O}(\sigma^4)$ which we see on the first three curves. However, if we integrate in time using a single fourth-order Runge–Kutta step, we see that our technique is now limited by the spatial discretization. The fourth series on the plot shows improvement over the first three, but convergence rate remains unchanged because this run uses a second-order spatial discretization. The fifth and sixth series use fourth- and



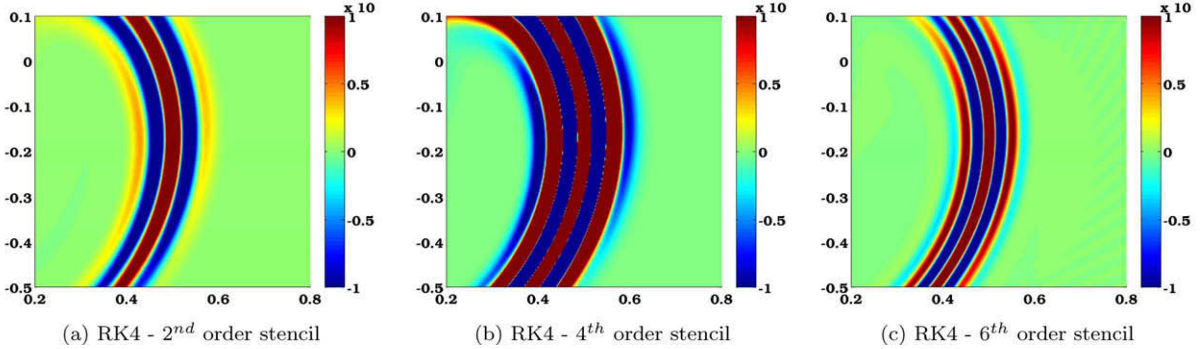
(a) FE - 2^{nd} order stencil

(b) RK4 - 2^{nd} order stencil

(c) RK4 - 6^{th} order stencil

(d) RK4 - 6^{th} order stencil

Fig. 13. Spatial behavior of the errors, relative to max(ω), in the reverse heat equation solutions using forward Euler (FE) or fourth-order Runge–Kutta (RK4) combined with a second- or sixth-order stencil for spatial derivatives. All panels have the same overlap ratio $\beta = 1$. Panels (a), (b) and (d) use a resolution of $\sigma = 4.102\ldots \times 10^{-3}$. Panel (c) uses a resolution of $\sigma = 3.281\ldots \times 10^{-2}$.



(a) RK4 - 2^{nd} order stencil

(b) RK4 - 4^{th} order stencil

(c) RK4 - 6^{th} order stencil

Fig. 14. Spatial behavior of the absolute errors in the reverse heat equation solutions using fourth order Runge–Kutta (RK4) with a second, fourth or sixth order stencil for spatial derivatives. All panels have the same overlap ratio $\beta = 1$ and resolution $\sigma = 4.102\ldots \times 10^{-3}$.

Table 3
Comparison of field interpolation schemes in vortex methods.

Method	Accuracy	Computational cost	Human effort	Notes
Estimate with ωh^2	Low	$\mathcal{O}(N)$ with known ω	Simple	Similar to Gaussian blurring
Spline interpolation of weights, e.g., M_4'	Moderate	Low storage, $\mathcal{O}(N)$ with known ω	Moderate	Interpolates circulations, not vorticity field values. Cannot specify initial conditions
RBF interpolation	Potentially high, with pre-conditioning	High storage unless matrix-free	Involved	Efficient implementations with $\mathcal{O}(N)$ operations possible
Reverse heat equation	Low ↔ high: depends on the method of integration	Low storage, $\mathcal{O}(N)$ with known ω	Moderate	Possesses all the computational advantages of ωh^2 but gains higher precision

sixth-order discretizations in space. Notice that the convergence rate jumps up in the fifth series because discretizations in time and space are both of sixth order. The sixth series is limited by the temporal integrator and hence has a similar convergence rate to the fifth series. The commonly-used ωh^2 method is also presented on the plot. The finite difference methods shown on the plot require a dozen or so multiplications and additions. Relative to the cost of the simulation itself, or the cost of calculating ω from a set of basis functions, the difference between using ωh^2 and using the reverse heat equation as a corrector is small, yet the improvement in accuracy is substantial.

To measure the impact of the overlap ratio β, we performed a study comparing $\beta = 1$ to $\beta = 2^{-1/2}$ as shown in Fig. 12. Reducing the overlap can have a substantial impact on linear solvers since it strengthens diagonal coupling, but for the explicit discretizations discussed here, it makes no difference on performance and only a slight difference in accuracy. As noted earlier, the overlap is often determined by other considerations such as the desired spatial accuracy of the particle method.

In Fig. 13(a) and (b), we can see the impact of accurate time integration on the size and structure of the error field. The five-banded spatial structure is indicative of the second derivative of the initial field in both cases, but the magnitude decreases by an order of magnitude. If we follow the rightmost column, panels Fig. 13(b) and (d), we see the results using the same time integrator, but panel Fig. 13(d) uses a sixth-order stencil for spatial derivatives indicated by the banded structure in the solution. If we follow along the bottom row, we can see the impact of spatial resolution on the quality of solutions created using the reverse heat equation. In panel Fig. 13(c), we have chosen a resolution of $\sigma = 4.640\ldots \times 10^{-2}$ where we can observe in Fig. 11 that the error curve is transitioning into its asymptotic regime at 6th order. The transition occurs at roughly the same value for both the fourth- and sixth-order stencils. Notice that the error in panel Fig. 13(d) is dominated by boundary effects, noted in the earlier discussion. Careful examination reveals that, as expected, the underresolved solution in panel Fig. 13(c) is not capable of capturing the fine spatial structure (all nine bands) that are present in the error field in panel Fig. 13(d). The five-, seven- and nine-banded structure in the error field can be hard to see while observing the full error field, so in Fig. 14 we use a more tightly focused color scaling, so that the banded oscillations stand out. The residual second, fourth and sixth derivatives are visible in the five-, seven- and nine-banded solutions.

5. Conclusions

We have studied the important process of field interpolation for Gaussian basis functions. Field interpolation is a general technique, but it plays a special role in the computational ecosystem surrounding vortex methods. Field interpolation is necessary for initialization of particles in flow simulations and for re-initializing distorted configurations of particles during long-time simulations. We have identified two techniques that provide results that are orders of magnitude more accurate than the common practice of estimation based on the field value at the collocation points, $\gamma_i = \omega_i h^2$. Of course, this estimation is simple to implement and imposes almost no additional computational overhead to a calculation, but the high-accuracy results generated by modern fourth-order elliptical vortex methods would be overwhelmed by low-order field interpolation methods. Thus, high-accuracy field interpolation techniques such as those introduced in this paper are necessary in this context. We summarize the available methods for field interpolation and the respective issues involved in Table 3.

Interpolation using radial basis functions (RBFs) provides a solution on a flexible geometry by solving a large linear system. We have addressed the ill-conditioning of RBF interpolation using a preconditioner based on a sparse approximation of the coefficient matrix, yielding substantial improvement in accuracy and algorithmic efficiency. The disadvantage is that one must still build and solve a large linear system. However, in another concurrent work, a fast algorithm to solve this system in $\mathcal{O}(N)$ operations has been developed and demonstrated in practice [35].

The reverse heat equation is a technique that provides systematic improvement to the ωh^2 estimate. The method is explicit, and requires low memory and computational effort. The accuracy depends upon the spatial and temporal discretization. The unique structure of this particular reverse heat equation method for solving field interpolation with Gaussians, which connects the duration of integration to the core width, allows us to circumvent the ill-posedness by taking a single step in time. The disadvantage of this particular technique is that the results are not as accurate as the RBF scheme, and while it is possible to perform the spatial derivatives on an irregular mesh, it adds a layer of complexity.

There are several clear avenues for extending this work. The most useful direction for future work is the development of an efficient algorithm so that given a field $\omega(x_i)$ measured at a set of points and a required overlap ratio β, one can determine an appropriate value of σ to provide a sufficiently accurate field interpolation. Also, we observe mesh effects in the error structure of the RBF collocation technique, but we have not identified an optimal arrangement given a field ω. One of the many advantages of RBF methods is that they offer flexibility in particle placement, and exploiting this feature would be a natural extension of this work. In this paper, we have limited the reverse heat equation solver to one temporal step to avoid amplification of high frequency noise, leaving open an exploration of the optimal number of permissible time steps given ω and field projection error requirements. As an extension of this work, we are interested in enhancing the fourth-order elliptical vortex method to allow multi-scale resolution in the fluid domain. In this sense, the RBF approach has the capability of dealing with particles of diverse size. What remains an important topic of research is how to analyze a flow simulation and (preferably on-the-fly) detect the relevant scales that need to be resolved, then adapt the particle sizes accordingly in different regions of the domain. We are currently investigating means of providing such a multi-resolution capacity, and devising accurate field interpolation when the basis functions have diverse sizes.

Acknowledgments

LAB acknowledges support from EPSRC (Engineering and Physical Sciences Research Council, UK) under Grant contract EP/E033083/1, and additional travel funds from Boston University College of Engineering. The authors thank the anonymous referees for their helpful comments and suggestions.

References

[1] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge University, 1994.
[2] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M. Knepley, L. Curfman-McInnes, B.F. Smith, H. Zhang. PETSc user's manual, Technical Report ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2002.
[3] L.A. Barba, Vortex method for computing high-Reynolds number flows: increased accuracy with a fully mesh-less formulation. Ph.D. Thesis, California Institute of Technology, 2004.
[4] L.A. Barba, A. Leonard, C.B. Allen, Numerical investigations on the accuracy of the vortex method with and without remeshing, in: 16th AIAA Computational Fluid Dynamics Conference, Number AIAA 2003-3426, American Institute of Aeronautics and Astronautics, 2003.
[5] L.A. Barba, A. Leonard, C.B. Allen, Advances in viscous vortex methods – meshless spatial adaption based on radial basis function interpolation, Int. J. Numer. Meth. Fluid 47 (5) (2005) 387–421.
[6] J.T. Beale, On the accuracy of vortex methods at large time, in: B. Engquist et al. (Eds.), Computational Fluid Dynamics and Reacting Gas Flows, Springer-Verlag, New York, 1988, pp. 19–32.
[7] J.T. Beale, A. Majda, Vortex methods I: convergence in three dimensions, Math. Comput. 39 (1982) 1–27.
[8] M.D. Buhmann, Radial Basis Functions. Theory and Implementations, Cambridge University Press, 2003.
[9] R.H. Chan, T.F. Chan, C.K. Wong, Cosine transform-based preconditioners for total variation deblurring, IEEE Trans. Image Proc. 8 (10) (1999) 1472–1478.
[10] P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, P. Koumoutsakos, Billion vortex particle direct numerical simulations of aircraft wakes, Comput. Meth. Appl. Mech. Eng. 197 (13–16) (2008) 1296–1304.
[11] J.P. Choquin, B. Lucquin-Desreux, Accuracy of a deterministic particle method for Navier–Stokes equations, Int. J. Numer. Meth. Fluid 8 (1988) 1439–1458.
[12] A.J. Chorin, Numerical study of slightly viscous flow, J. Fluid Mech. 57 (1973) 785–796.
[13] G.-H. Cottet, P. Koumoutsakos, Vortex Methods. Theory and Practice, Cambridge University Press, 2000.
[14] G.E. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific, 2007.
[15] D. Firsov, S.H. Lui, A fast deblurring algorithm, Appl. Math. Comput. 183 (2006) 285–291.
[16] J. Goodman, T.Y. Hou, J. Lowengrub, Convergence of the point vortex method for the 2D Euler equations, Commun. Pure Appl. Math. XLIII (1990) 415–430.
[17] P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, J. Comput. Phys. 138 (1997) 821–857.
[18] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.
[19] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (1980) 289–335.
[20] A. Leonard, Computing three-dimensional incompressible flows with vortex elements, Ann. Rev. Fluid Mech. 17 (1985) 523–559.
[21] A. Leonard, D. Shiels, J.K. Salmon, G.S. Winckelmans, P. Ploumhans, Recent advances in high-resolution vortex methods for incompressible flows, AIAA 97-2108, 1997.
[22] M.V. Melander, J.C. McWilliams, N.J. Zabusky, Axisymmetrization and vorticity-gradient intensification of an isolated two-dimensional vortex through filamentation, J. Fluid Mech. 178 (1987) 137–159.
[23] J. Monaghan, Smoothed particle hydrodynamics, Ann. Rev. Astron. Astrophys. 30 (1992) 543–574.
[24] J.J. Monaghan, Extrapolating B-splines for interpolation, J. Comput. Phys. 60 (1985) 253–262.
[25] W.B. Muniz, F.M. Ramos, H.F. de Campos Velho, Entropy- and Tikhonov-based regularization techniques applied to the backwards heat equation, Int. J. Comput. Math. 40 (2000) 1071–1084.
[26] S. Osher, L.L. Rudin, Feature-oriented image enhancement using shock filters, SIAM J. Numer. Anal. 27 (4) (1990) 929–940.
[27] R.B. Platte, L.F. Rossi, T.B. Mitchell, Using global interpolation to evaluate the Biot–Savart integral for deformable elliptical Gaussian vortex elements, SIAM J. Sci. Comput. 31 (3) (2009) 2342–2360.
[28] L. Rosenhead, The formation of vortices from a surface of discontinuity, Proc. Roy. Soc. Lond. A A134 (1931) 170–192.
[29] L.F. Rossi, Resurrecting core spreading vortex methods: a new scheme that is both deterministic and convergent, SIAM J. Sci. Comput. 17 (1996) 370–397.
[30] L.F. Rossi, Merging computational elements in vortex simulations, SIAM J. Sci. Comput. 18 (1997) 1014–1027.
[31] L.F. Rossi, Achieving high-order convergence rates with deforming basis functions, SIAM J. Sci. Comput. 26 (3) (2005) 885–906.
[32] L.F. Rossi, A comparative study of Lagrangian methods using axisymmetric and deforming blobs, SIAM J. Sci. Comput. 27 (4) (2006) 1168–1180.
[33] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Co., Boston, 1996.
[34] T.I. Seidman, Optimal filtering for the backward heat equation, SIAM J. Numer. Anal. 33 (1) (1996) 162–170.
[35] C.E. Torres, L.A. Barba, Fast radial basis function interpolation with Gaussians by localization and iteration, J. Comput. Phys. 228 (2009) 4976–4999, doi:10.1016/j.jcp.2009.03.007.