Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 213 (2006) 239–263
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Modified interpolation kernels for treating diffusion and remeshing in vortex methods

Daehyun Wee, Ahmed F. Ghoniem *

Massachusetts Institute of Technology, Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA

Received 10 March 2005; received in revised form 10 August 2005; accepted 11 August 2005
Available online 23 September 2005

Abstract

A scheme treating diffusion and remeshing, simultaneously, in Lagrangian vortex methods is proposed. The vorticity redistribution method is adopted to derive appropriate interpolation kernels similar to those used for remeshing in inviscid methods. These new interpolation kernels incorporate diffusion as well as remeshing. During implementation, viscous splitting is employed. The flow field is updated in two fractional steps, where the vortex elements are first convected according to the local velocity, and then their vorticity is diffused and redistributed over a predefined mesh using the extended interpolation kernels. The error characteristics and stability properties of the interpolation kernels are investigated using Fourier analysis. Numerical examples are provided to demonstrate that the scheme can be successfully applied in complex problems, including cases of nonlinear diffusion.

Keywords: Numerical simulation; Computational particle methods; Vortex methods; Redistribution; Diffusion

1. Introduction

Lagrangian vortex methods [6,25] are tools for computing complex fluid flows. Several of the computational advantages of these methods are:

(1) While Eulerian methods introduce extra dispersion or dissipation, even in flows with zero velocity gradient, such errors are minimized during advection in Lagrangian vortex methods.
(2) The condition of numerical stability is not restricted by the CFL condition.
(3) The support of particle distribution remains a small fraction of the total volume of the flow field, determined by where vorticity is confined. The method is endowed with natural 'grid adaptivity', and hence the computational elements are utilized more efficiently.

* Corresponding author. Tel.: +1 617 2532295; fax: +1 617 2535981.
E-mail address: ghoniem@mit.edu (A.F. Ghoniem).

0021-9991/$ - see front matter © 2005 Published by Elsevier Inc.
doi:10.1016/j.jcp.2005.08.009

(4) The method provides a natural way to represent small vortical structures that arise at high Reynolds numbers.

While Lagrangian vortex methods were originally formulated for inviscid flows, successful approaches for viscous flows have been proposed [5,8,9,11,29,31]. In some methods, such as random walk [5] and diffusion velocity methods [11], particles are transported while their strength remains fixed. In other methods, the strength assigned to each particle is allowed to change without displacing the particles. In many cases, more particles are introduced to capture the expanding region where vorticity is confined.

One popular algorithm is the PSE (particle strength exchange) scheme [9], in which the diffusion equation is converted into integro-differential equations, which are discretized in space by approximating the integral using a quadrature rule. The semi-discrete equations are again discretized in time in various different ways—implicitly or explicitly—up to whatever order of accuracy is desired. This method has been successfully applied to several complex flows [19,27,39,41], and has been extended to the case of anisotropic diffusion [10], and to the case with spatially variable radius of the cutoff function [7].

The use of a quadrature rule in PSE requires relatively uniform particle distribution, and this naturally necessitates frequent remeshing. Remeshing is also implemented in other methods, even in inviscid simulations to satisfy other conditions. For instance, it has been observed that long-time accuracy of convection computation deteriorates severely due to the distortion of the particle distribution [6,14]. Several local regridding schemes have been devised to solve this problem, by inserting new particles where inter-particle distance becomes too large [16,17,41]. These schemes are limited to geometrically simple flows, and tend to grow the number of particles rapidly, unless careful clustering and merging is also implemented. For these reasons, global remeshing is now considered necessary in most Lagrangian particle methods, and the design and verification of various remeshing schemes have become an active research area [1,3].

In this article, we design a scheme that treats diffusion and remeshing simultaneously and without additional ambiguity or computational overhead. The scheme, 'redistribution onto a grid', will be formulated as an extension of the vorticity redistribution method [33], and cast in the form of interpolation kernels, which resemble those used in inviscid remeshing [6,18].

The paper is organized as follows. In Section 2, the vorticity redistribution method is introduced. Next, we develop the modified interpolation kernels in Section 3. The error characteristics and the stability properties of these kernels are investigated in Section 4. We finally provide numerical examples in Section 5.

2. The redistribution method

The vorticity redistribution method, or simply the redistribution method, developed in [33] is a deterministic approach to solve the constant-diffusivity diffusion equation. In this method, the fundamental solution of the diffusion equation for each particle vorticity is approximated by a new set of particles within a ball of a finite radius, whose locations and strengths are determined by satisfying a number of 'predictive moment matching conditions'. The latter enforce the requirement that the vorticity assigned to the new particles have approximately the same moments, up to a certain order, as the moments of the fundamental solution generated by the source particle. The new particle vorticity is obtained by redistributing the source particle strength onto the target particles, i.e., by transferring fractions of the source particle strength to the target particles nearby. The spatial resolution of the method is naturally defined by the redistribution radius, that is, the radius of the ball in which the target particles for each source particle lie.

How to obtain a redistribution formula that determines the correct redistribution fractions that satisfy the predictive moment matching conditions depends on the specific problem of interest. When the fundamental solution of the diffusion equation is known explicitly, the moments of the fundamental solution can be exactly determined, and the corresponding redistribution formula can be easily constructed [33]. However, for spatially varying or anisotropic diffusion, the explicit form of the fundamental solution is often not available. To address this difficulty, a more general method to design redistribution formulae satisfying the moment matching conditions was proposed [13,32], in which the evolution equations for the moments of the fundamental solution of each source particle were discretized by explicit integration schemes,

such as the forward Euler scheme. The redistribution formulae were obtained by applying the particle approximation to evaluate the resulting integrals. This method, referred as the Galerkin formulation [21], is more general, and hence we briefly describe it in this section.

Consider the one-dimensional heat equation with spatially dependent conductivity, $v(x)$, as in [32]:

$$\mathscr{L}u \equiv \frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(v(x)\frac{\partial u}{\partial x}\right) = 0, \tag{1}$$

where u is the temperature, and the spatial variable is x. We assume that $v(x)$ is positive and its pointwise value and derivatives up to the second order are uniformly bounded. $\mathscr{L}$ has a fundamental solution $Z(x, \xi, t, \tau)$ that satisfies the equation

$$\mathscr{L}(x, t, \partial_x, \partial_t)Z(x, \xi, t, \tau) = \delta(x - \xi)\delta(t - \tau). \tag{2}$$

The particle approximation consists of the discrete sum of δ distributions, i.e.,

$$\underline{u}^n = \sum_{i=1}^{N} \Gamma_i^n \delta(x - x_i^n), \tag{3}$$

where x_i^n denotes the location of the ith particle at the nth time step, and Γ_i^n is its strength. At $t = 0$, $\Gamma_i^0 = u(x_i^0)\Delta x$, where x_i^0 is the initial location of the ith particle. The initial locations of the particles are assumed to be distributed over the support of the initial temperature field with an equal spacing Δx. At each time step, for each source particle, we define the kth moment of the fundamental solution, $G_{k,i}^n$, and its approximation, $\underline{G}_{k,i}^n$, as follows:

$$G_{k,i}^n = \int_{\mathbf{R}^d} (x - x_i^{n-1})^{(k)} Z(x, x_i^{n-1}, \Delta t_{\mathrm{d}}, 0)\mathrm{d}x, \tag{4}$$

and

$$\underline{G}_{k,i}^n = \int_{\mathbf{R}^d} \left(\sum_{j=1}^{N} f_{ij}^n \delta(x - x_j^n)\right)(x - x_i^{n-1})^{(k)}\,\mathrm{d}x = \sum_{j=1}^{N} f_{ij}^n (x_j^n - x_i^{n-1})^{(k)}. \tag{5}$$

f_{ij}^n is the redistribution fraction, that is, the fraction of the strength of the ith particle transferred to the jth particle at the nth time step. We use standard notations, i.e., $x^{(k)} = x_1^{k_1} x_2^{k_2} x_3^{k_3} \cdots x_d^{k_d}$, and $|k| = k_1 + k_2 + k_3 + \cdots + k_d$, where d is the dimension of the space. In this one-dimensional problem, $d = 1$, and hence, $x^{(k)} = x^k$, and $|k| = k$. Δt_{d} is the time step.

Next, multiplying (1) by $(x - x_i^{n-1})^k$ and integrating by parts, the evolution equations for $G_{k,i}^n$, where $0 \leqslant k \leqslant 2$, can be obtained:

$$\begin{aligned}
\frac{\mathrm{d}G_{0,i}^n}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbf{R}} Z(x, x_i^{n-1}, t, 0)\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial Z}{\partial t}\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial}{\partial x}\left(v(x)\frac{\partial Z}{\partial x}\right)\mathrm{d}x = 0,\\
\frac{\mathrm{d}G_{1,i}^n}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbf{R}} Z(x, x_i^{n-1}, t, 0)(x - x_i^{n-1})\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial Z}{\partial t}(x - x_i^{n-1})\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial}{\partial x}\left(v(x)\frac{\partial Z}{\partial x}\right)(x - x_i^{n-1})\mathrm{d}x\\
&= \int_{\mathbf{R}} Z(x, x_i^{n-1}, t, 0)\frac{\mathrm{d}v}{\mathrm{d}x}\mathrm{d}x,\\
\frac{\mathrm{d}G_{2,i}^n}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbf{R}} Z(x, x_i^{n-1}, t, 0)(x - x_i^{n-1})^2\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial Z}{\partial t}(x - x_i^{n-1})^2\mathrm{d}x = \int_{\mathbf{R}} \frac{\partial}{\partial x}\left(v(x)\frac{\partial Z}{\partial x}\right)(x - x_i^{n-1})^2\mathrm{d}x\\
&= 2\int_{\mathbf{R}} Z(x, x_i^{n-1}, t, 0)\left(v(x) + (x - x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right)\mathrm{d}x.
\end{aligned} \tag{6}$$

The redistribution formulae are designed by discretizing these equations using the forward Euler scheme and utilizing the following expression for particle distribution:

$$\begin{aligned}\underline{u}^{n-1} &= \sum_{i=1}^{N} \Gamma_i^{n-1}\delta(x - x_i^{n-1}),\\ \underline{u}^{n} &= \sum_{i=1}^{N} \Gamma_i^{n}\delta(x - x_i^{n}) = \sum_{i=1}^{N} \Gamma_i^{n-1}\left(\sum_{j=1}^{N} f_{ij}^{n}\delta(x - x_j^{n})\right).\end{aligned} \tag{7}$$

The outcomes are the discrete equations describing the evolution of $\underline{G}_{k,i}^{n}$ expressed in terms of f_{ij}^{n}:

$$\begin{aligned}\frac{\underline{G}_{0,i}^{n} - \underline{G}_{0,i}^{n-1}}{\Delta t_{\mathrm{d}}} &= \frac{1}{\Delta t_{\mathrm{d}}}\left(\sum_j f_{ij}^{n} - 1\right) = 0,\\ \frac{\underline{G}_{1,i}^{n} - \underline{G}_{1,i}^{n-1}}{\Delta t_{\mathrm{d}}} &= \frac{1}{\Delta t_{\mathrm{d}}}\sum_j f_{ij}^{n}(x_j^{n} - x_i^{n-1}) = \int_{\mathbf{R}} Z(x, x_i^{n-1}, 0, 0)\frac{\mathrm{d}v}{\mathrm{d}x}\mathrm{d}x = \left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)_{x=x_i^{n-1}},\\ \frac{\underline{G}_{2,i}^{n} - \underline{G}_{2,i}^{n-1}}{\Delta t_{\mathrm{d}}} &= \frac{1}{\Delta t_{\mathrm{d}}}\sum_j f_{ij}^{n}(x_j^{n} - x_i^{n-1})^2 = 2\int_{\mathbf{R}} Z(x, x_i^{n-1}, 0, 0)\left(v(x) + (x - x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right)\mathrm{d}x = 2v(x_i^{n-1}).\end{aligned} \tag{8}$$

Therefore, the corresponding redistribution formulae for f_{ij}^{n} are

$$\begin{aligned}&\sum_j f_{ij}^{n} = 1,\\ &\sum_j f_{ij}^{n}(x_j^{n} - x_i^{n-1}) = \left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)_{x=x_i^{n-1}}\Delta t_{\mathrm{d}},\\ &\sum_j f_{ij}^{n}(x_j^{n} - x_i^{n-1})^2 = 2v(x_i^{n-1})\Delta t_{\mathrm{d}}.\end{aligned} \tag{9}$$

Assuming that the redistribution radius scales as $\mathrm{O}(\Delta t_{\mathrm{d}}^{1/2})$, it can be shown that the global truncation error of scheme (9) behaves as $\mathrm{O}(\Delta t_{\mathrm{d}}^{1/2-\epsilon})$, where ϵ is a small positive real number, as shown in Appendix A. The procedure works in the same way for higher-order spatial accuracy [13]. The redistribution formulae (9) reduce to those in [33] for the case of constant diffusivity.

3. Redistribution onto a uniform grid—the modified interpolation kernels

To determine f_{ij}^{n} from the given redistribution formulae, we need to specify the target particle locations x_i^{n}. In the original treatment [33], the neighboring particles of each source particle were chosen as the target particles, i.e., the set of x_i^{n-1} and that of x_i^{n} were taken to be the same. More particles were introduced if the number of neighboring particles was not sufficient to achieve the desired accuracy. Although this approach makes the entire process grid-free, the complex procedure necessary to deal with the arbitrariness of the number and the locations of the target particles makes the original redistribution method expensive, especially in three-dimensional simulations where the number of particles easily reaches several millions. In this section, we provide an alternative formulation to address this difficulty.

From the formulation described in the previous section, the following fact can be easily recognized: it is not necessary to keep the same particle locations before and after each redistribution step, i.e., the set of x_i^{n-1} and that of x_i^{n} in (7) need not be the same. For example, we can simply take a set of uniform grid points as the target particle locations to develop a redistribution formula for each source particle. In that case, the arbitrariness in the number and locations of target particles is eliminated, and the complex procedure of finding the fractions is replaced by a much simpler one.

Since multi-dimensional generalization is straightforward, we consider the one-dimensional case first. We concentrate on the case of constant diffusivity first. Suppose we have an equally spaced grid, located at $x = -\Delta x$, 0, and Δx. We interpret these grid points as the target particle locations, i.e., $x_1 = -\Delta x$, $x_2 = 0$, and $x_3 = \Delta x$. Given that a source particle is located at $x = x_0$, where $|x_0| < \frac{\Delta x}{2}$, the corresponding redistribution formula, to the lowest order in $\mathbf{R}$, is given by

$$\sum_{j=1}^{3} f_{0j} = 1,$$
$$\sum_{j=1}^{3} f_{0j}(x_j - x_0) = 0, \tag{10}$$
$$\sum_{j=1}^{3} f_{0j}(x_j - x_0)^2 = 2\nu\Delta t_{\mathrm{d}}.$$

Solving these equations explicitly, we obtain the following redistribution fractions:

$$f_{01} = \frac{2\nu\Delta t_{\mathrm{d}} - x_0\Delta x + x_0^2}{2\Delta x^2},$$
$$f_{02} = \frac{\Delta x^2 - 2\nu\Delta t_{\mathrm{d}} - x_0^2}{\Delta x^2}, \tag{11}$$
$$f_{03} = \frac{2\nu\Delta t_{\mathrm{d}} + x_0\Delta x + x_0^2}{2\Delta x^2}.$$

Although these expressions are enough for implementation, rewriting them in the form of an interpolation kernel is more convenient for further discussion. For a given particle distribution $\underline{u}^n$ in the form of (3), we define the interpolated particle distribution $\underline{u}^{n+1}$ as

$$\underline{u}^{n+1}(x) = \sum_{j\in\mathbf{Z}} \delta(x - j\Delta x) \int_{\mathbf{R}} \lambda\left(\frac{j\Delta x - x'}{\Delta x}\right)\underline{u}^n(x')\mathrm{d}x' = \sum_{i=1}^{N} \Gamma_i^n \left(\sum_{j\in\mathbf{Z}} \lambda\left(\frac{j\Delta x - x_i^n}{\Delta x}\right)\delta(x - j\Delta x)\right). \tag{12}$$

The interpolated particle distribution has particles only at $x = j\Delta x$, where $j \in \mathbf{Z}$. We call λ the interpolation kernel, since it relates the initial particle distribution and the interpolated distribution. Usually λ is of compact support. Thus the interpolation of a particle is only done over its nearest grid points. Since (12) exhibits some similarity to (7), we can easily convert (11) to the following interpolation kernel:

$$\Lambda_2(\xi, c) = \begin{cases} \frac{1}{2}(1 - |\xi|)(2 - |\xi|) + c^2 & \frac{1}{2} \leqslant |\xi| < \frac{3}{2}, \\ 1 - |\xi|^2 - 2c^2 & |\xi| < \frac{1}{2}, \\ 0 & \frac{3}{2} \leqslant |\xi|, \end{cases} \tag{13}$$

where $c = \sqrt{\nu\Delta t_{\mathrm{d}}}/\Delta x$. The corresponding redistribution formulae approximate the diffusion process with a global truncation error $\mathrm{O}(h)$, where $h = \sqrt{\Delta t_{\mathrm{d}}}$ [33], if c is kept constant during refinement. The notation Λ_2 has been chosen intentionally. This expression yields one of the classical 'inviscid' interpolation kernels given in [6] at the limit of $c \to 0$, where it was also denoted as Λ_2. One may realize that Λ_2 becomes the TSC (triangular-shaped cloud) interpolation kernel when $c^2 = 1/8$. This fact can be used to estimate the effective kinematic viscosity induced by the numerical diffusion when one uses the TSC interpolation kernel for remeshing.

The procedure given above can be generalized to other kernels. Two of the most widely used interpolation kernels, Λ_3 and M_4', can also be extended to account for diffusion as follows:

$$\Lambda_3(\xi, c) = \begin{cases} 1 - 2c^2 + |\xi|\left(3c^2 - \frac{1}{2}\right) - \xi^2 + \frac{|\xi|^3}{2} & |\xi| < 1, \\ (2 - |\xi|)\left(\frac{1}{6}(3 - |\xi|)(1 - |\xi|) + c^2\right) & 1 \leqslant |\xi| < 2, \\ 0 & 2 \leqslant |\xi| \end{cases} \tag{14}$$

and

$$M_4'(\xi, c) = \begin{cases} 1 - \frac{5\xi^2}{2} + \frac{3|\xi|^3}{2} - c^2(2 - 9\xi^2 + 6|\xi|^3) & |\xi| < 1, \\ \frac{1}{2}(2 - |\xi|)^2(1 - |\xi| - 2c^2 + 4c^2|\xi|) & 1 \leqslant |\xi| < 2, \\ 0 & 2 \leqslant |\xi|. \end{cases} \tag{15}$$

Λ_3 is continuous, and $M'_4 \in C^1(\mathbf{R})$. Λ_3 approximates the diffusion process with a global truncation error $O(h^2)$, and M'_4, $O(h)$. When $c^2 = 1/6$, these two kernels coincide.

So far, we have discussed the one-dimensional cases only. The multi-dimensional generalization of these interpolation kernels can be achieved in a trivial way for a uniform Cartesian grid. One can obtain redistribution fractions in $\mathbf{R}^d$ simply by using the tensor product of the redistribution fractions obtained by the interpolation kernel acting on each coordinate. The resulting redistribution fractions automatically satisfy the redistribution formulae given in [33].

The idea of redistributing particle strength onto uniform grid points can also be applied to the case of variable diffusivity. Here, we present the interpolation kernel satisfying (9)

$$\Lambda'_3(\xi, c, d_v) = \begin{cases} \frac{1}{6}(2-\xi)(6c^2 + 3d_v(2-\xi) + (1-\xi)(3-\xi)) & 1 \leqslant \xi < 2, \\ \frac{1}{2}(2 + 2c^2(3\xi - 2) - \xi(1 - \xi(\xi - 2) - d_v(4 - 3\xi))) & 0 \leqslant \xi < 1, \\ \frac{1}{2}(2 - 2c^2(3\xi + 2) + \xi(1 - \xi(\xi + 2) + d_v(4 + 3\xi))) & -1 \leqslant \xi < 0, \\ \frac{1}{6}(2+\xi)(6c^2 - 3d_v(2+\xi) + (1+\xi)(3+\xi)) & -2 < \xi < -1, \\ 0 & 2 \leqslant |\xi|, \end{cases} \tag{16}$$

where $c^2 = \nu\Delta t_d/\Delta x^2$ and $d_v = \frac{\Delta t_d}{\Delta x}\frac{\partial v}{\partial x}$. v and $\frac{\partial v}{\partial x}$ should be evaluated at the source particle location. Because only three equations are available in (9), while there are four unknown redistribution fractions, we imposed an additional condition for the third-order moment, i.e., $\sum_j f_{ij}^n (x_j^n - x_i^{n-1})^3 = d_v \Delta x^3$. The resulting kernel is continuous, and approximates the diffusion process with a global truncation error $O(h^{M'})$ for all $M' < 1$, as shown in Appendix A. As the notation implies, this expression gives Λ_3 in (14) when $d_v = 0$.

Again, multi-dimensional generalization can be made simply by taking tensor products. The procedure of multi-dimensional generalization gives redistribution formulae which are different from those obtained directly from the Galerkin formulation given in [13]. For instance, in $\mathbf{R}^2$, for $k_1 = k_2 = 1$, $\underline{G}^n_{k,i}$ is $O(h^4)$ if it is obtained by taking tensor products. On the other hand, according to [13], $\underline{G}^n_{k,i}$ must be exactly zero under the same condition. However, as one can clearly see in this example, the difference only contributes at a higher order than the error considered, hence these two different formulae are equivalent within the error considered.

The actual implementation of these interpolation kernels to simulate diffusion in vortex methods is a straightforward generalization of the original redistribution method [33]. To solve the Navier–Stokes equation in the velocity–vorticity formulation, we employ the viscous splitting algorithm [6,25]: the evolution of the flow field is considered in discrete time steps. In each step, the vortex elements are first convected, and then diffused by interpolation, i.e., the algorithm consists of substeps where the convective and the diffusive effects are considered separately. In this way, the computational advantages of Lagrangian vortex methods, that is, minimal dispersion/dissipation during the computation of convection, no restriction from the CFL condition, and optimal utilization of computational elements, are automatically inherited without being compromised, because convection is still dealt with in completely Lagrangian way.

It is convenient to define appropriate notations for different step sizes, because the time step for diffusion is often chosen as a multiple of that for convection at high Reynolds number. Thus, from here on, the convection time step size is denoted by Δt_c, while the diffusion time step size is denoted by Δt_d. If there is no need to distinguish between different time steps, as in Appendix A, we use Δt as Δt_d.

Due to the core overlap condition imposed during the convection substep, the grid size for interpolation, Δx, should be chosen such that $\Delta x < \sigma$, where σ is the radius of the cutoff function. The choice of Δt_d and Δx is further restricted by the stability bound on c^2 associated with each interpolation kernel. These stability bounds will be discussed in Section 4. To meet all these conditions simultaneously, one may first decide on σ by considering the spatial resolution required for the solution, and then decide on a value of Δx that satisfies the overlap condition. After that, Δt_d can be chosen as a multiple of Δt_c in the range of valid values for Δt_d, which should be decided by stability consideration.

We note that the use of these interpolation kernels for treating diffusion has the following advantages. First, the use of a uniform grid eliminates the expensive linear optimization process used to find the fraction in the original redistribution scheme, and results in a very efficient diffusion scheme. The high computational load resulting from the optimization process was one of the most critical weaknesses of the original vorticity redistribution method [6]. The second is its simplicity. An inviscid vortex code can be expanded easily to treat

viscous flows. If the code already has a routine for remeshing onto a uniform grid, simply modifying the kernel leads to a viscous flow code. Finally, the two processes, remeshing and diffusion, are treated in one step in such a way that the dispersive errors introduced by remeshing are controlled by the concomitant diffusion process, and provides an easy way to guarantee the stability of remeshing.

On the other hand, there are several potential weaknesses. The application of the method might require the generation of a large data set for the storage of the grid points, as in other implementations of global remeshing. However, only the grid points near the support of the particle distribution are relevant. If the support of particle distribution is large, one may still avoid the problem of generating a large data array by partitioning particles into several small clusters and performing interpolation for each cluster separately. An elegant strategy of tree-structured grid storage is available [42], and can be easily adopted for the current scheme. The order of approximation of the interpolation kernels presented here is relatively low. Λ_3, which has the highest order among the interpolation kernels given, is first-order in time and second-order in space. However, constructing higher-order interpolation kernels is possible, though we do not pursue it in this paper.

We end this section with few comments concerning the relation between the method proposed here and other diffusion simulation approaches. The first is the finite difference method. One may treat diffusion and remeshing by first performing remeshing through an inviscid interpolation kernel, then by applying an explicit finite difference scheme on the remeshed particle distribution. Such a two-step approach is valid, but there are differences between this and our one-step approach. The two-step approach does not in general yield particle distribution identical to that obtained by the modified interpolation kernels. For instance, suppose that we have only one source particle initially. If one first applies the inviscid Λ_3 to this particle and then uses the three-point centered finite difference formula in space and the explicit Euler scheme in time to treat diffusion, the support of resulting particle distribution covers six grid points in general, which is larger than that covered by the modified Λ_3. If one uses the four-point one-sided finite difference formulae for the outermost remeshed particles to limit the support of resulting particle distribution, some of moment conditions are violated. Actually, Λ_3 given here represents the only particle distribution covering four points with all the moments up to the third-order correct. Our one-step approach usually results in more efficient utilization of grid points.

We also note that a similar idea of using a grid to simulate diffusion was proposed in [24]. However, this early treatment was based on the concept of resampling [6], i.e., the redistribution fraction onto each grid point is determined by the local value of the fundamental solution, not by matching the moments. Since each moment corresponds to an integral property such as the total circulation, the current scheme has better conservation properties. For example, the current method preserves the linear impulse in the case of constant diffusivity, where the method in [24] cannot. Finally, we note that the use of a quadrature rule and the nature of semi-discretization make it conceptually difficult to incorporate the idea of remeshing within PSE directly. This is one reason why the discussion has been made on the basis of the redistribution method.

4. Error analysis of interpolation kernels

In this section, we analyze the error characteristics of the extended interpolation kernels presented in the previous section. The purpose of this analysis is to obtain the stability bound of each kernel. We discuss the dissipative or dispersive characteristics of the error for the low-frequency modes. Since the high-frequency modes are all well damped. We also show that the dispersive nature of these interpolation kernels is changed by the addition of diffusion.

To this end, we consider the one-dimensional linear advection–diffusion equation with constant flow speed U, i.e.,

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}. \tag{17}$$

If we employ a typical operator splitting algorithm, the advection step is solved by particle methods without introducing any additional error at each time step. One can simply discretize the initial condition using particles, and displace the particles to obtain the field at any time instance. The error is introduced during the diffusion step, or equivalently during the remeshing step, by the application of an interpolation kernel.

Let Δt be the time step for interpolation and Δx be the grid size. We denote $x_j = j\Delta x$ as the fixed location of the jth grid point and $\underline{u}_j^n$ as the strength of the particle located at x_j immediately following the nth remeshing step. We assume for convenience that $0 < C \equiv \frac{U\Delta t}{\Delta x} < \frac{1}{2}$. Since interpolation is made onto the nearest grid points of the source particle, $U\Delta t$ can be arbitrarily large. This restriction makes it possible to interpret the resulting evolution equation of $\underline{u}_j$ as an Eulerian scheme, making the analysis easier. The results obtained from this analysis are not affected by the removal of this restriction, since we can always use a transform $x' = x - nt\Delta x/\Delta t$, where n is chosen such that the flow speed measured in the new coordinate system satisfies the restriction.

We first consider Λ_2. Since particles are always remeshed onto the uniform grid points at the end of the step, the position of the jth particle at the beginning of a new time step is x_j. In the advection step, the particles are displaced by $U\Delta t$, i.e., $\tilde{x}_j^n = x_j + U\Delta t$, and $\tilde{u}_j^n = \underline{u}_j^n$, where $\tilde{x}_j$ is the location of the displaced jth particle, and $\tilde{u}_j$ is its strength. In the remeshing step, the displaced particles are interpolated onto the uniform grid points. Thus, we have

$$\underline{u}_j^{n+1} = \underline{u}_{j-1}^n \Lambda_2\left(\frac{x_j - \tilde{x}_{j-1}^n}{\Delta x}, c\right) + \underline{u}_j^n \Lambda_2\left(\frac{x_j - \tilde{x}_j^n}{\Delta x}, c\right) + \underline{u}_{j+1}^n \Lambda_2\left(\frac{x_j - \tilde{x}_{j+1}^n}{\Delta x}, c\right). \tag{18}$$

Again, $c = \frac{\sqrt{\nu\Delta t}}{\Delta x}$. This formula actually gives the Lax–Wendroff scheme when $c^2 = 0$.

To analyze the error characteristics from this expression, one usually calculates the amplification factor and the phase speed error [4]. The analysis is performed in the wave number space. We take the Fourier transform of (18) using the following substitution:

$$\underline{u}_j^n = \sum_\theta \underline{v}_\theta^n \mathrm{e}^{\mathrm{i}j\theta}, \tag{19}$$

where $\mathrm{i} = \sqrt{-1}$, and $\theta = 2\pi k\Delta x$. Reorganizing (18) by using this substitution, we can find

$$\underline{v}_\theta^{n+1} = g(C, c, \theta)\underline{v}_\theta^n, \tag{20}$$

where

$$g(C, c, \theta) = 1 - c^2 - C^2 + \left(c^2 + \frac{C^2}{2} + \frac{C}{2}\right)\mathrm{e}^{-\mathrm{i}\theta} + \left(c^2 + \frac{C^2}{2} - \frac{C}{2}\right)\mathrm{e}^{\mathrm{i}\theta}. \tag{21}$$

From this amplification factor, we first obtain the stability bound. In practice, particles can be placed anywhere, and hence we do not have any control over C and θ. Thus, we need to obtain the range of c where $|g(C, c, \theta)| \leqslant 1$ for all $0 \leqslant C \leqslant \frac{1}{2}$ and $0 \leqslant \theta \leqslant 2\pi$. Either analytically or numerically, we can compute $|g(C, c, \theta)|^2$ to obtain this range of valid c. For Λ_2, the range of valid c is $c^2 \leqslant 3/8$. Within this range, the l^2 norm of the discretized field variable is decreased by the application of Λ_2.

To analyze the error characteristics of the low-frequency modes, we compare the effect of dispersion and that of dissipation, in an order of magnitude sense. The exact solution in the wave number space is given by

$$v_\theta^{n+1} = \mathrm{e}^{-\mathrm{i}2\pi kU\Delta t - 4\pi^2\nu k^2\Delta t}\underline{v}_\theta^n = \mathrm{e}^{-\mathrm{i}C\theta - c^2\theta^2}\underline{v}_\theta^n. \tag{22}$$

This implies that the rate of norm decay and the phase speed error should be analyzed in the following way:

$$g(C, c, \theta) = \mathrm{e}^{-\mathrm{i}C\theta + \mathrm{i}A(C,c,\theta) - c^2\theta^2 - B(C,c,\theta)}, \tag{23}$$

where $A(C, c, \theta)$ represents the phase speed error and hence the dispersive effect, and $B(C, c, \theta)$ represents the rate of additional norm decay and hence the effect of numerical dissipation. It is hard to get exact expressions of $A(C, c, \theta)$ and $B(C, c, \theta)$, but we can get the following asymptotic formulae for the leading order terms by taking the logarithm of (21) and expanding it in series

$$\begin{aligned} A(C, c, \theta) &= \frac{1}{6}C(1 - 6c^2 - C^2)\theta^3 + \mathrm{O}(\theta^5), \\ B(C, c, \theta) &= \frac{1}{24}\left(12c^4 - 3C^2(C^2 - 1) - 2c^2(1 + 6C^2)\right)\theta^4 + \mathrm{O}(\theta^6). \end{aligned} \tag{24}$$

This analysis shows that the leading order error induced by the application of Λ_2 is strictly dominated by its dispersive component at the low-frequency regimes, i.e., as $\theta \to 0$. When $c^2 \leqslant 1/8$, the leading order term of $A(C, c, \theta)$ is greater than or equal to 0 for all $0 \leqslant C \leqslant \frac{1}{2}$ and $0 \leqslant \theta \leqslant 2\pi$. Thus these low-frequency modes show lagging phase error, and induce spurious oscillations at the trailing edges of the wave packets [38,40]. Especially when c^2 is very small, these unphysical oscillations can survive dissipation for a time long enough to couple with other parts of the equations in more complex equations, e.g., the Euler equation or the Navier–Stokes equation at high Reynolds number.

If the equation of interest is extremely sensitive to such spurious oscillations, one may completely suppress these oscillations by enforcing the monotonicity preservation condition [22,40]. To achieve this, one can simply adjust c^2 to make these interpolation kernels non-negative by choosing Δx and Δt_d appropriately. For Λ_2, when $1/8 \leqslant c^2 \leqslant 3/8$, the interpolation kernel is guaranteed to be non-negative everywhere. A non-negative interpolation kernel gives non-negative redistribution fractions, and hence makes the resulting scheme TVD, which guarantees the preservation of monotonicity [22,40]. However, since these interpolation kernels have been successfully used with $c^2 = 0$ in many previous inviscid vortex simulations [18], we conclude that the use of non-negative interpolation kernels may not lead to serious instability in vortex simulations. In other applications, there are still possibilities that these spurious oscillations may be troublesome.

It is interesting to see that the dispersion relation can be significantly modified by the addition of diffusion: especially when $c^2 > 1/8$, the low-frequency modes may show leading phase error for certain values of C. This threshold actually coincides with the lower bound of the range of c^2 yielding Λ_2 non-negative.

For Λ_3 and M_4', we only state the results briefly. For Λ_3, the amplification factor is given by

$$\begin{aligned} g(C,c,\theta) = {} & C\left(c^2+\frac{1}{6}(C-1)(C+1)\right)e^{-2i\theta} + \left(c^2 + C - 3c^2C + \frac{C^2}{2} - \frac{C^3}{2}\right)e^{-i\theta} + 1 \\ & - 2c^2 - \frac{C}{2} + 3c^2C - C^2 + \frac{C^3}{2} + (1-C)\left(c^2 - \frac{1}{6}(2-C)C\right)e^{i\theta}. \end{aligned} \tag{25}$$

When $c^2 \leqslant 1/2$, $|g(C, c, \theta)| \leqslant 1$ for all $0 \leqslant C \leqslant \frac{1}{2}$ and $0 \leqslant \theta \leqslant 2\pi$. This gives the stability bound as $c^2 \leqslant 1/2$. We can also get the asymptotic expressions for $A(C, c, \theta)$ and $B(C, c, \theta)$.

$$\begin{aligned} A(C,c,\theta) &= \frac{1}{60}C(2 + C - C^2 - 10c^2)(1 - 3C + 2C^2)\theta^5 + O(\theta^7), \\ B(C,c,\theta) &= \frac{1}{24}(12c^4 - c^2(2 + 12C - 12C^2) + C(2 - C - 2C^2 + C^3))\theta^4 + O(\theta^6). \end{aligned} \tag{26}$$

For small values of c^2, the leading order term in $A(C, c, \theta)$ is greater than or equal to 0, and the application of Λ_3 may also result in spurious oscillations at the trailing edge. To suppress these oscillations completely, one can choose $1/6 \leqslant c^2 \leqslant 1/2$ to make Λ_3 non-negative.

For M_4', we get

$$\begin{aligned} g(C,c,\theta) = {} & \left(3c^2C^2 - 2c^2C^3 - \frac{C^2}{2} + \frac{C^3}{2}\right)e^{-2i\theta} + \left(c^2 + \frac{C}{2} + 2C^2 - 9c^2C^2 - \frac{3}{2}C^3 + 6c^2C^3\right)e^{-i\theta} \\ & + 1 - 2c^2 - \frac{5C^2}{2} + 9c^2C^2 - 6c^2C^3 + \frac{3C^3}{2} + \left(c^2 - \frac{C}{2} + C^2 - 3c^2C^2 - \frac{C^3}{2} + 2c^2C^3\right)e^{i\theta}. \end{aligned} \tag{27}$$

The stability bound is given by $c^2 \leqslant 1/2$. The asymptotic expressions for $A(C, c, \theta)$ and $B(C, c, \theta)$ are given by

$$\begin{aligned} A(C,c,\theta) &= \frac{1}{6}C(1 - 6c^2)(1 - 3C + 2C^2)\theta^3 + O(\theta^5), \\ B(C,c,\theta) &= \frac{1}{24}(12c^4 - 2c^2(1 + 24C^2 - 48C^3 + 24C^4) + 9C^2(C-1)^2)\theta^4 + O(\theta^6). \end{aligned} \tag{28}$$

Again, the leading order term in $A(C, c, \theta)$ is greater than or equal to 0 for $c^2 \leqslant 1/6$, and hence the low-frequency modes show lagging phase error. Interestingly, when $c^2 > 1/6$, $A(C, c, \theta)$ becomes non-positive for any C, and the low-frequency modes show leading phase error. The non-negativity of M_4' can be achieved

by choosing $1/6 \leqslant c^2 \leqslant 1/2$. Again, the threshold of the radical change of the dispersion relation for low-frequency modes coincides with the lower bound of c^2, yielding non-negativity.

For Λ_3', it is hard to get a precise stability bound, since there is another parameter d_v. Numerical calculation of the upper bound of $|g(C, c, \theta)|$ shows that $|d_v|$ should remain small when c^2 is small. However, for moderate values of c^2, the restriction on d_v is not severe. We also note that d_v approaches 0 if one refines the resolution while keeping c^2 constant. Thus, for most cases, one can just check the bound on c^2, for which one may consult the case of Λ_3. The region of non-negativity for Λ_3' is a convex set in (c^2, d_v) plane, whose boundary is described by the following set of equations:

$$\begin{aligned} &c^2 - \frac{1}{2} = 0, \quad 4 + 9d_v^2 - 24c^2 = 0, \\ &20 + 27|d_v| - 162c^2|d_v| + 54|d_v|^3 - 2(7 - 18c^2 + 9d_v^2)^{3/2} = 0. \end{aligned} \tag{29}$$

5. Numerical examples

In this section, we discuss the results of a number of computations using the modified interpolation kernels. We provide three numerical examples of three-dimensional vortex simulations. Next, a nonlinear reaction–diffusion problem is solved to demonstrate the applicability of the method to cases of spatially varying diffusion coefficient.

5.1. Vortex rings

The behavior of vortex rings has been studied intensively, and hence they serve as good examples to check the capability of a numerical method [34]. We show results of three examples: evolution of a vortex ring at an intermediate Reynolds number, asymptotic drift of a vortex ring, and a case of side-by-side collision of two vortex rings.

We first briefly discuss the numerical method. A viscous splitting algorithm is employed. During the convection step, we solve the equations of motion for inviscid incompressible flow in vorticity transport form:

$$\begin{aligned} &\frac{\mathrm{D}\omega}{\mathrm{D}t} = \omega \cdot \nabla v, \\ &\nabla \cdot v = 0, \end{aligned} \tag{30}$$

where $\omega = \nabla \times v$, and v is the velocity. The numerical solution proceeds by discretizing the vorticity field onto overlapping vector elements, each centered at χ_i^c with volume $\mathrm{d}V_i$ and vorticity ω_i

$$\omega(x,t) = \sum_i^N [\omega_i \mathrm{d}V_i](t)\phi_\sigma(x - \chi_i^c(t)). \tag{31}$$

The vorticity associated with each element is localized by a radially symmetric cutoff function ϕ_σ of radius σ, where $\phi_\sigma(x) = \frac{1}{\sigma^3}\phi(\frac{|x|}{\sigma})$. We use the low-order algebraic kernel as the cutoff function [23,41]. Each vortex element is described by a 'stick', decomposing the particle strength $[\omega_i \mathrm{d}V_i](t)$ into a positive scalar weight Γ_i times a material line element $\delta\chi_i(t)$. The vector $\delta\chi_i$ points in the direction of the vorticity, and is ascribed to two nodes. Nodes are simply advected by the velocity field:

$$\frac{\mathrm{d}\chi_i}{\mathrm{d}t} = v(\chi_i). \tag{32}$$

Advecting the nodes accounts for the material line element deformation, and thus for stretching and tilting of the vorticity $\omega_i \mathrm{d}V_i$. A second-order predictor/corrector scheme with adaptive time-step control is used for time integration of the ordinary differential equations in (32), where the velocity at each node $v(\chi_i)$ is evaluated by an adaptive tree-code [23]. When $|\delta\chi_i|$ of a given element exceeds 0.9σ, a new node is added halfway between the original two nodes. The parallel implementation of the adaptive tree-code is achieved by domain decomposition using the k-means clustering technique [26].

During the diffusion step, we use Λ_3 to interpolate the particle strength $[\omega_i \mathrm{d}V_i]$ of each vortex element onto target particles on a uniform grid. Following the interpolation, we eliminate particles with $|\omega_i \mathrm{d}V_i| < |\omega \mathrm{d}V|_{\mathrm{del}}$ to control the problem size. Next, each target particle on the grid is converted back into a stick, having its center χ_i^c at the grid point and $|\delta\chi_i| = 0.6\sigma$. Because $|\delta\chi_i|$ defines the length scale at which ∇v is evaluated for the calculation of stretching, one should choose $|\delta\chi_i|$ comparable to σ, which defines the spatial resolution of the simulation. If $|\delta\chi_i|$ is too small, stretching is evaluated at a length scale that is not well resolved. On the other hand, $|\delta\chi_i|$ should not be too large to avoid a quick increase in number of elements, since we add a new node when $|\delta\chi_i| > 0.9\sigma$. Our $|\delta\chi_i|$ is chosen via a tuning process considering these conditions. After the conversion of the target particles into sticks, the code can start the convection step again using these sticks as its initial condition.

All of the following simulations were performed on the IBM SP-RS/6000 located at the National Energy Research Scientific Computing Center (NERSC).

5.1.1. Evolution of a vortex ring at an intermediate Reynolds number

The first example is a single vortex ring at $Re_\Gamma = \Gamma/\nu = 500$. A ring of radius R and core radius a is initially placed at the $y = 0$ plane. The core of the ring is represented as

$$\omega_\phi = \frac{K}{\pi}\frac{\Gamma}{a^2}\exp\left\{-K\left(\frac{R^2}{a^2}+\frac{r^2}{a^2}-\frac{2Rr}{a^2}\sin\theta\right)\right\}, \tag{33}$$

where $r = \sqrt{x^2+y^2+z^2}$, $\tan\theta = (\sqrt{x^2+z^2})/y$, and $K = (2.24182)^2/4$. To make the initial distribution smooth, an image ring was placed across the axis of symmetry so that $\omega_\phi = 0$ at the y-axis. The ring has unit circulation and unit radius, i.e., $\Gamma = 1$ and $R = 1$. The core radius is chosen to be $a/R = 0.35$. This set of parameters makes the initial conditions identical to those of the axisymmetric spectral simulation performed by Stanaway et al. [35,36].

We performed two different runs. The first is a fully three-dimensional simulation at $\Delta t_c = \Delta t_d = 0.5$, $\sigma = 0.1$, $\Delta x = 0.07$, $|\omega \mathrm{d}V|_{\mathrm{del}} = 10^{-8}$. In the second case, a 20° section of the vortex ring is simulated at higher spatial resolution, where $\Delta t_c = \Delta t_d = 0.25$, $\sigma = 0.05$, $\Delta x = 0.035$, $|\omega \mathrm{d}V|_{\mathrm{del}} = 10^{-10}$. If an element lies outside the 20° section, the element is rotated into the domain using azimuthal symmetry. Since the fully three-dimensional run did show a symmetry in the azimuthal direction during the period of interest, the simulation using the 20° section is expected to behave similarly.

The results are reported in the following dimensionless variables, which were also used in [35,36]. The dimensionless speed of the vortex ring centroid is given by

$$\overline{U} = U_c\frac{(I_0/\rho)^{1/2}}{\nu^{3/2}}, \tag{34}$$

where I_0 is the initial linear impulse of the ring, and U_c is the speed measured in the computational units. We also use dimensionless time, which is scaled as

$$\overline{t} = t\frac{\nu^2}{I_0/\rho}, \tag{35}$$

and shifted to match the initial time reported in [35,36].

In Fig. 1, the speed of the vortex ring centroid is plotted. For comparison, the curve reported in [35,36] is also shown. The values are underestimated in the lower resolution run, but the higher resolution run shows a close match. At the later stage, where diffusion plays a dominant role in establishing the vorticity distribution, close agreement is observed at both resolutions. The circulation of the vortex ring is plotted in Fig. 2. Unlike the initial speed, which is more strongly affected by convection than diffusion, the evolution of the circulation is well captured even by the lower resolution run.

We also show vorticity contours of the high resolution run on the $z = 0$ plane in Fig. 3. We have chosen the same instances as those reported in [35,36] for one-to-one comparison. The contour levels remain the same for all times in this figure. To recover the contour levels used in [35,36], we have matched the diameter of the outermost solid contour at the initial condition, and the difference between the solid lines is set to be a factor of 10

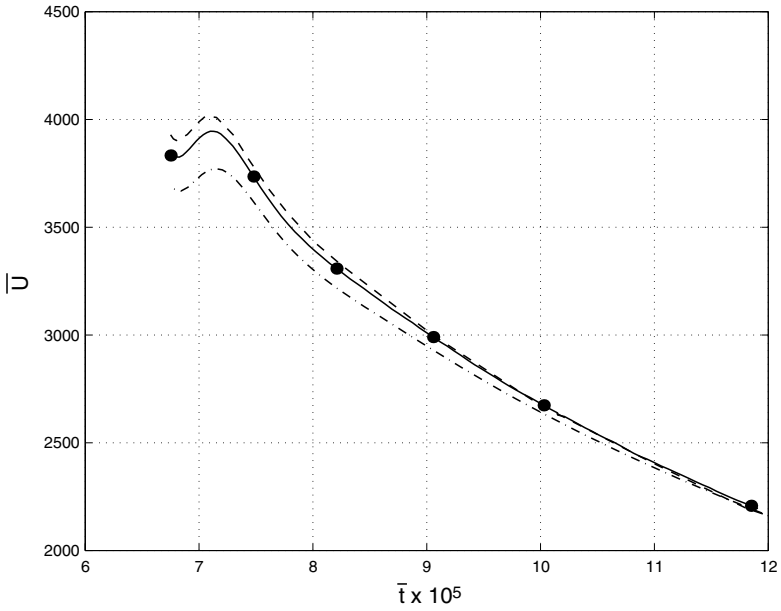

Fig. 1. Speed of the vortex ring verses time. Solid, 20° section simulation at high resolution; dash-dot, full simulation at low resolution; dashed, Stanaway et al. [35,36]. Dots on the solid curve correspond to the instances shown in Fig. 3.

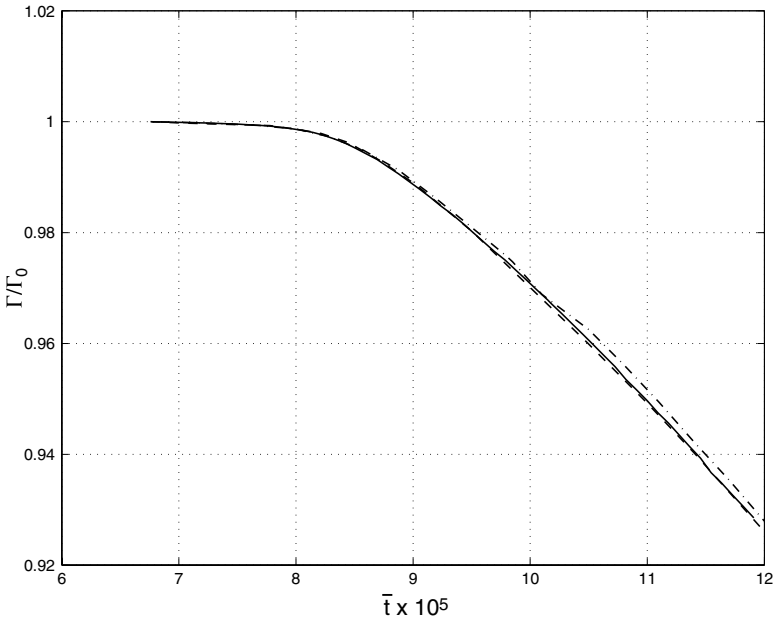

Fig. 2. Circulation of the vortex ring verses time. Solid, 20° section simulation at high resolution; dash-dot, full simulation at low resolution; dashed, Stanaway et al. [35,36].

larger than between the dashed ones. Comparing Fig. 3 to that reported in [35,36], we see that our simulation does reproduce the details of the vortex ring correctly. Even the subtle structure of the tail is well matched. We also note that the linear impulse of the ring is preserved within 0.9% for the duration of the simulation in both of our simulations. The number of vortex elements at the end of the simulation was around 500,000 for both of our simulations.

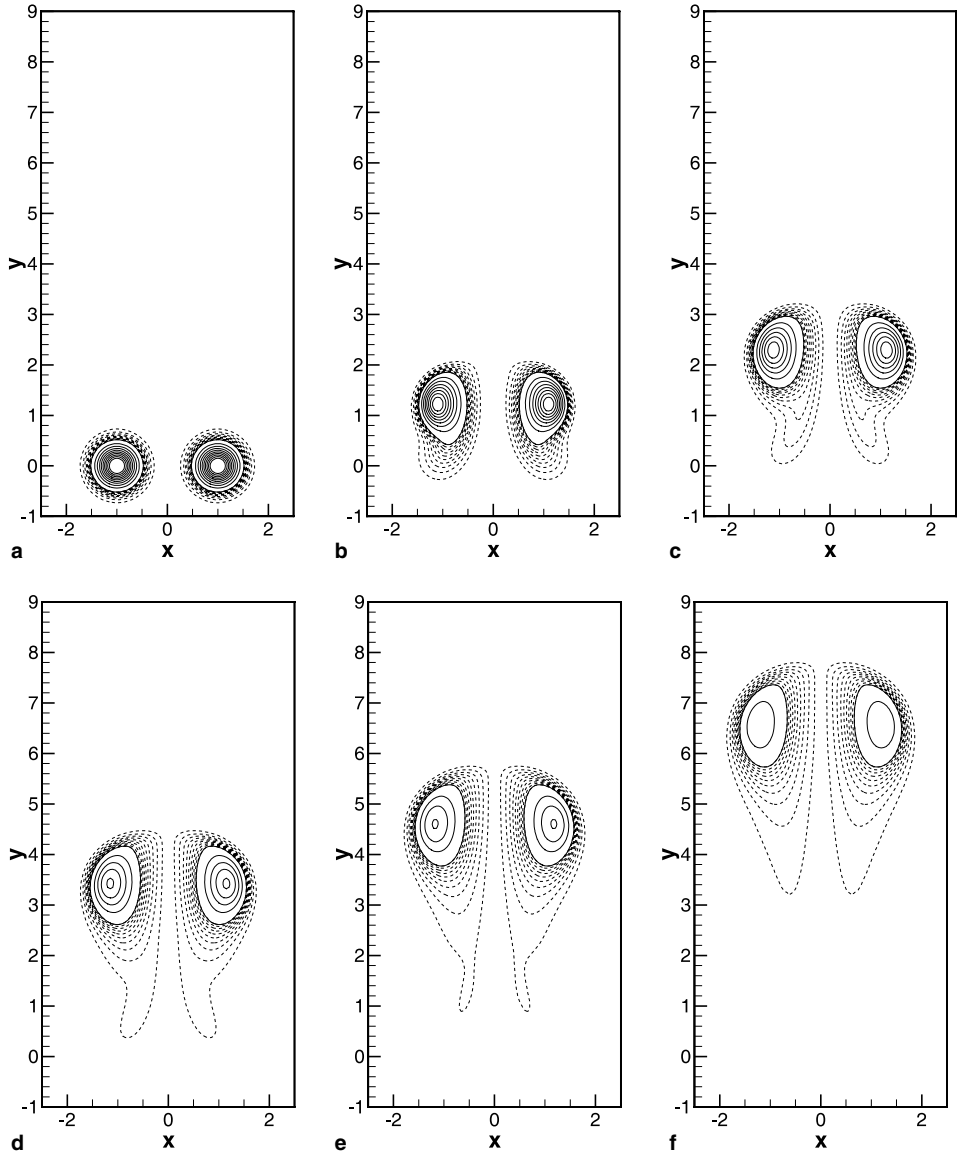

Fig. 3. The evolution of a single vortex ring. Vorticity contour at several instants in time. The contour levels for dashed lines vary from $|\omega_z| = 0.024$ to $|\omega_z| = 0.24$. The contour levels for solid lines vary from $|\omega_z| = 0.24$ to $|\omega_z| = 2.4$. For lines of the same type, the vorticity varies linearly. (a) $\bar{t} = 6.75 \times 10^{-5}$, (b) $\bar{t} = 7.48 \times 10^{-5}$, (c) $\bar{t} = 8.21 \times 10^{-5}$, (d) $\bar{t} = 9.06 \times 10^{-5}$, (e) $\bar{t} = 10.03 \times 10^{-5}$ and (f) $\bar{t} = 11.85 \times 10^{-5}$.

5.1.2. Asymptotic drift of a vortex ring

The second example is the long-time asymptotic drift of a vortex ring. The centroid speed under these conditions was studied by Rott and Cantwell [30], and we compare the result of our simulation to these theoretical estimates.

Initially, we place a Stokes vortex ring at the $y = 0$ plane. The vorticity distribution of a Stokes vortex ring is given by

$$\omega_\phi = \frac{I_0/\rho}{8\pi^{3/2}(\nu t_1)^2} \sin\theta\eta \exp(\eta^2), \tag{36}$$

where $\eta = r/\sqrt{4\nu t_1}$. r and θ are defined in the same way as in the previous example. The ring has unit linear impulse, i.e., $I_0/\rho = 1$, and the kinematic viscosity is chosen to be $\nu = 1$ for simplicity. With this choice, the

only parameter that must be specified is the initial time t_I, which is chosen to be $t_I = 1/900$. The initial Reynolds number is $Re_I = (I_0/\rho)^{1/2}/(\nu t_I)^{1/2} = 30$. The numerical parameters used are: $\Delta t_c = \Delta t_d = 0.0003$, $\sigma = 0.05$, $\Delta x = 0.03$, $|\omega \mathrm{d}V|_{del} = 10^{-12}$. To limit the size of the simulation, we again follow the evolution of a 20° section of the ring, assuming azimuthal symmetry. The 20° section of the Stokes vortex ring is initially discretized into more than 80,000 elements, which gives smaller inter-particle distance than that specified by Δx. Still, the initial centroid speed of the vortex ring is naturally underestimated, since the numerical parameters, such as σ, are chosen to match the resolution required for the later stage. Note that our purpose is to study the long-time asymptotic drift, where diffusion is expected to dominate the dynamics.

The speed of the long-time asymptotic drift of a single vortex ring is given as follows [30]:

$$\overline{U} = \frac{7}{15}(8\pi\bar{t})^{-3/2} \approx 0.0037038\bar{t}^{-3/2}. \tag{37}$$

This theoretical result was also well verified by the axisymmetric simulations of Stanaway et al. [35,36]. As shown in Fig. 4, where the speed of the vortex ring centroid is plotted, the result of our simulation matches (37) well as $\bar{t}$ increases.

Figs. 5 and 6 show the evolution of the circulation and that of the kinetic energy respectively. From the expression of the Stokes vortex ring, it can be shown that the circulation must evolve as $\bar{t}^{-1}$, and that the kinetic energy must evolve as $\bar{t}^{-3/2}$. Our simulation matches these trends exactly. The linear impulse of the ring is preserved within 0.04 % for the duration of the simulation. The error in the linear impulse increases mostly at the initial stage, where convection still affects the evolution of the vortex ring. At the later stage, where diffusion dominates the evolution, the error in the linear impulse does not increase much, showing that interpolation indeed preserves the linear impulse. The number of vortex elements at the end of the simulation was around 300,000.

5.1.3. Side-by-side collision of two vortex rings

The final example of vortex calculations is the interaction of two vortex rings; the case studied by Kida et al. [15]. As an initial condition, two identical vortex rings are placed side-by-side. The centers of the vortex rings are placed on the x-axis, separated by a distance s. The radius of each ring is R. We use a Gaussian vorticity distribution within the core

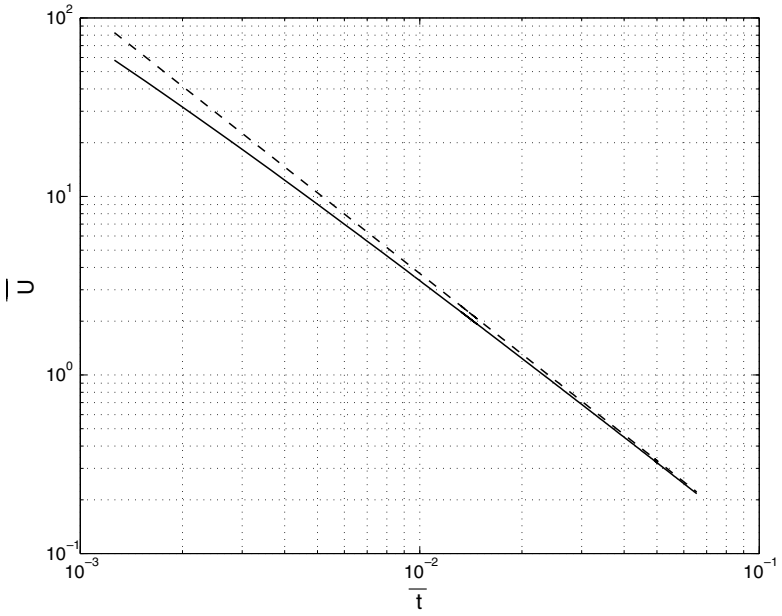

Fig. 4. Speed of the vortex ring verses time. Solid, present study; dashed, Eq. (37) [30].

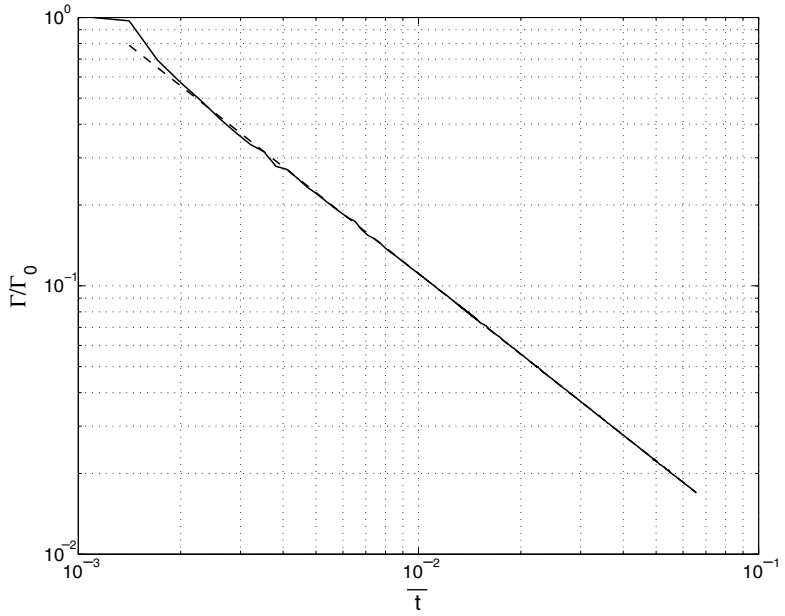

Fig. 5. Circulation of the vortex ring verses time. Solid, present study; dashed, a line proportional to $\overline{t}^{-1}$.

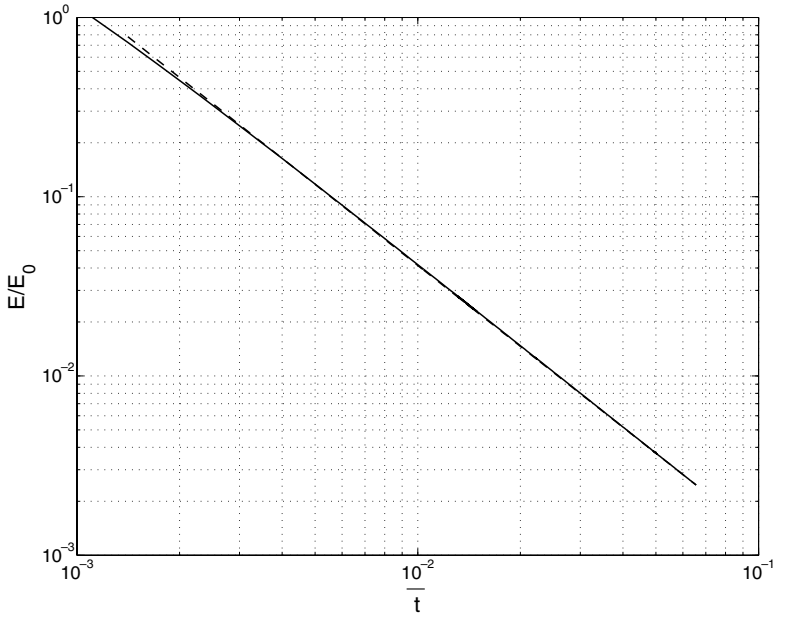

Fig. 6. Kinetic energy of the vortex ring verses time. Solid, present study; dashed, a line proportional to $\overline{t}^{-3/2}$.

$$\omega_\phi = \omega_0 \exp\left\{-\left(\frac{r}{a}\right)^2\right\}, \tag{38}$$

where r is the distance from the core centerline, ω_0 is the maximum vorticity at the core center. The nominal circulation of the vortex ring is $\pi\omega_0 a^2$. Note that (38) is equivalent to (33) with the proper change of variables. We use different representations to simplify the comparison with the reference cases.

A set of parameters similar to Case I in [15] is chosen, namely, $R = 0.982$, $a = 0.393$, $s = 3.65$, $\omega_0 = 23.8$, and $v = 0.01$, which makes $Re_\Gamma = 1153$ based on the nominal circulation. The rings are not inclined with respect to the $y = 0$ plane. This condition is not identical to that of Case I in [15], since the simulation performed in [15] was spatially periodic, while our rings are isolated in $\mathbf{R}^3$. Due to periodicity, the evolution of the vortex rings in [15] turned out to be slower. Thus the comparison between these two cases can be qualitative only.

The numerical parameters used are the following: $\Delta t_c = 0.05$, $\Delta t_d = 0.1$, $\sigma = 0.2$, $\Delta x = 0.090909$, and $|\omega \, dV|_{del} = 5 \times 10^{-7}$. The vortex rings move toward the $x = 0$ plane, as they travel in the y direction, through

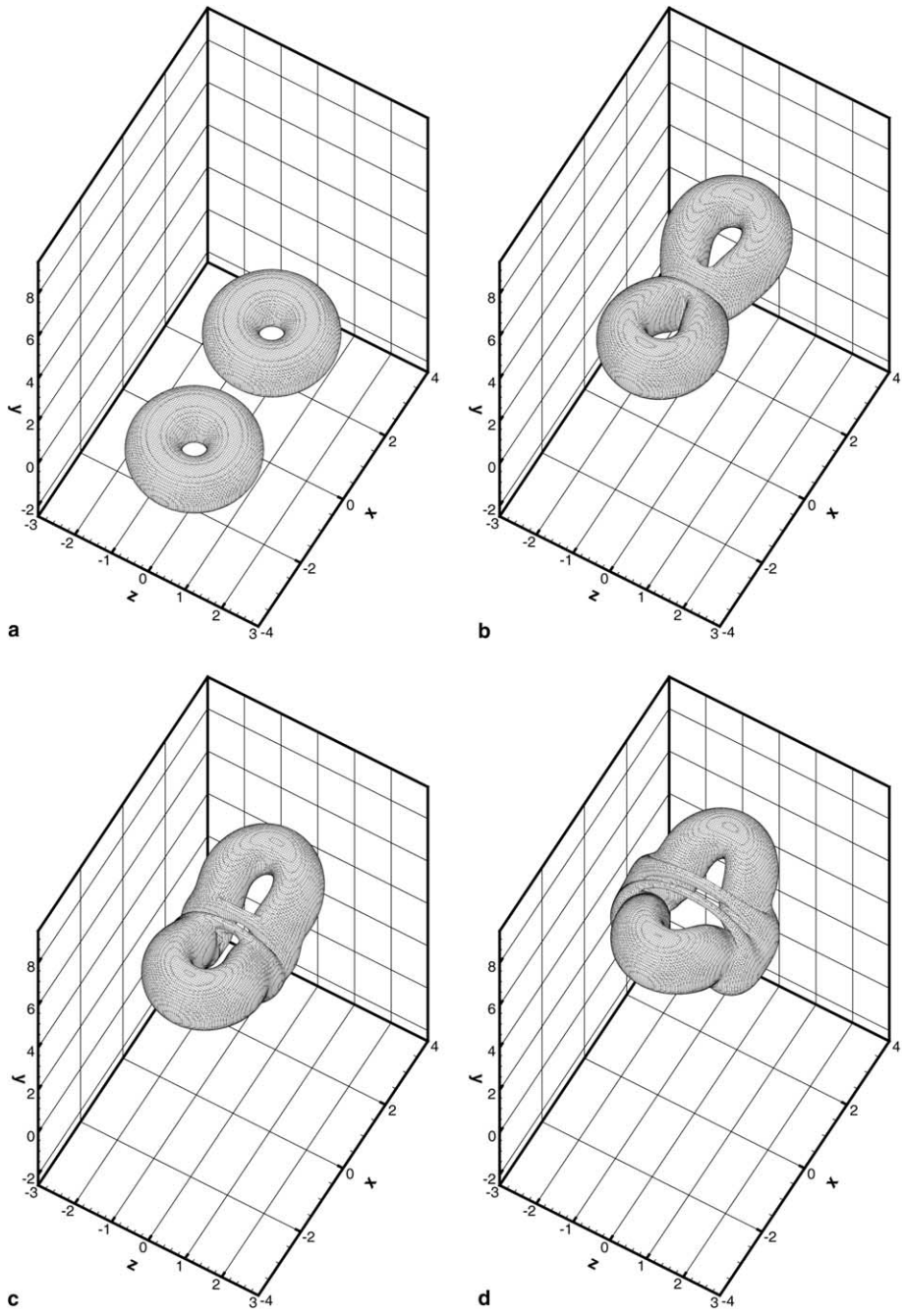

Fig. 7. Iso-surfaces of the vorticity norm $|\omega| = 2.0$. (a) $t = 0.0$, (b) $t = 3.0$, (c) $t = 4.0$ and (d) $t = 5.0$.

their mutual induction. They approach each other, and collide along the $x = 0$ plane as shown in Fig. 7. The collision promotes the establishment of large vorticity gradients, which are then gradually annihilated by diffusion. Eventually, the two rings merge into a distorted single ring. The evolution of colliding rings shown in Fig. 7 is similar to that depicted in [15]. In particular, one can recognize the formation of threads on the front of the vortex tube, which are remnants of the anti-parallel vortices at the contact point. These threads were also observed in [15].

The interaction can be seen in Figs. 8 and 9, where the contours of ω_z and ω_x are plotted, respectively. As depicted in Fig. 8, the outer cores move upward faster than the inner cores, and induce a flow that forces the inner cores toward the $x = 0$ plane. As the inner cores collide, the outer vortex tubes extend across the $x = 0$ plane resulting in the formation of bridges, as shown in Fig. 9. During this first reconnection, the circulation of each of the inner cores decreases rapidly, while the circulation of each of the bridges increases, which can be

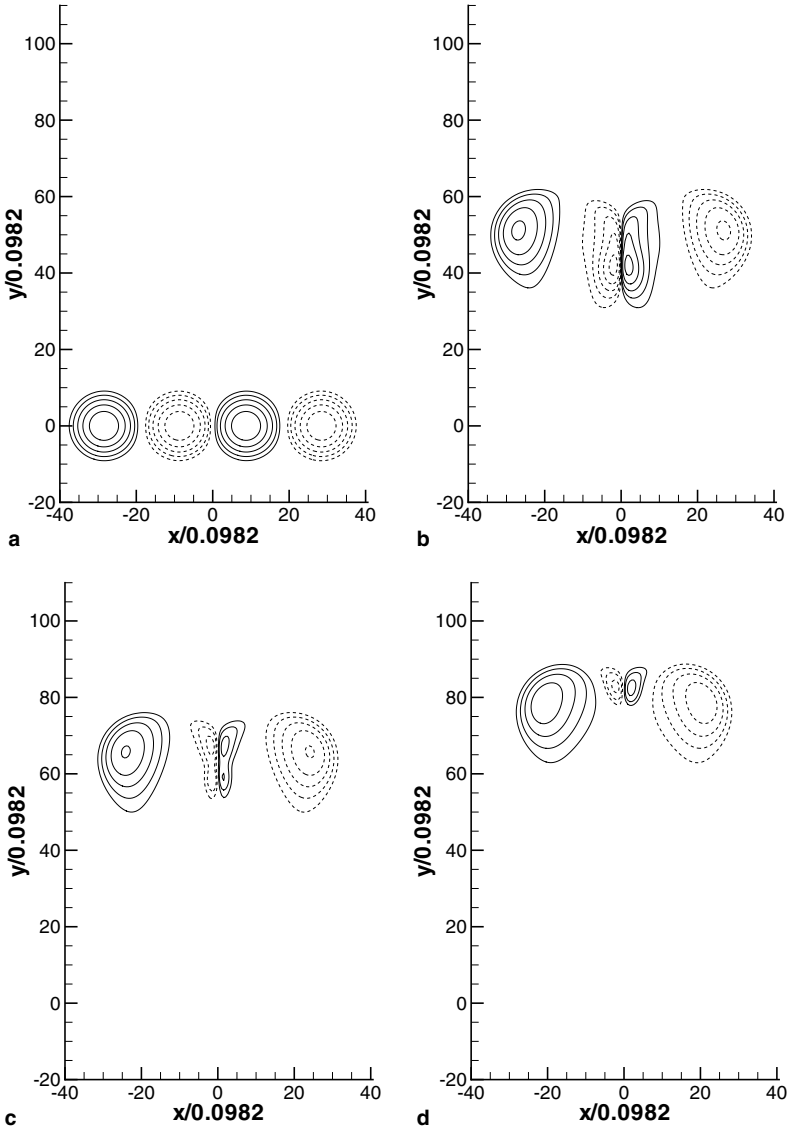

Fig. 8. Contour of ω_z at $z = 0$. Levels plotted are 0.5, 1, 2, 4, and 8. Solid and dashed lines represent positive and negative values respectively. (a) $t = 0.0$, (b) $t = 3.0$, (c) $t = 4.0$ and (d) $t = 5.0$.

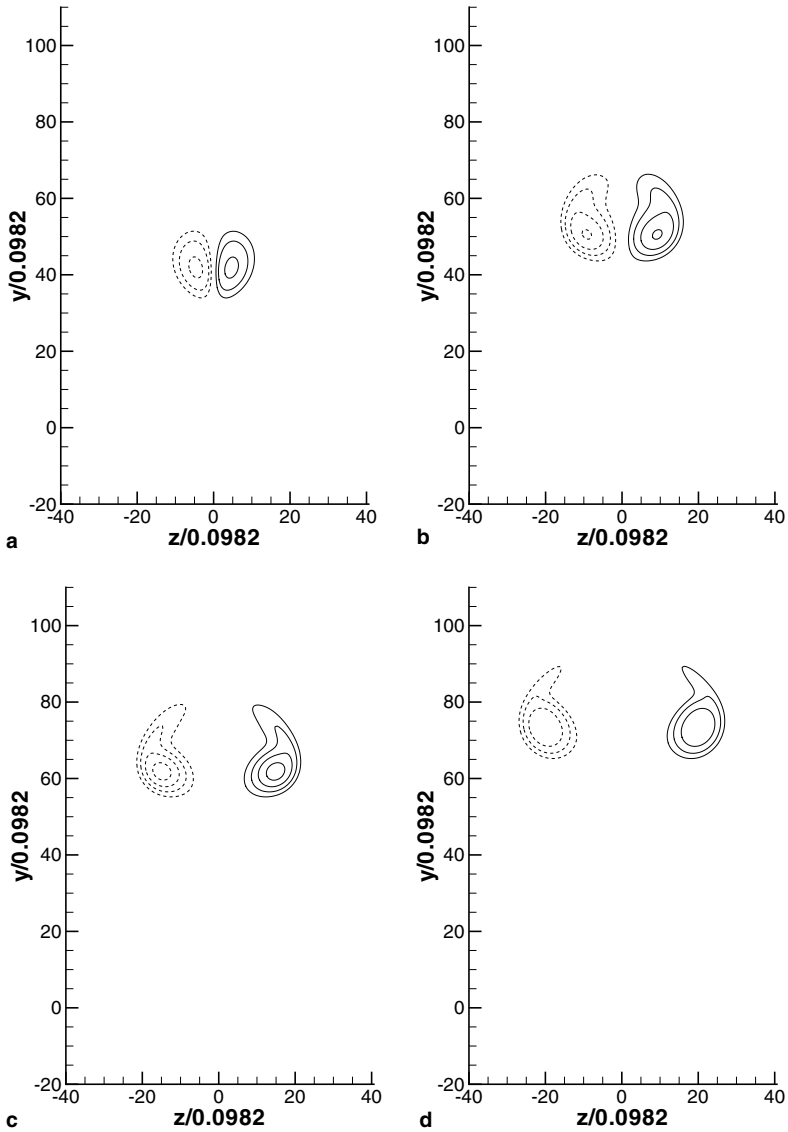

Fig. 9. Contour of ω_x at $x = 0$. Levels plotted are 0.5, 1, 2, 4, and 8. Solid and dashed lines represent positive and negative values respectively. (a) $t = 3.0$, (b) $t = 4.0$, (c) $t = 5.0$ and (d) $t = 6.0$.

seen in Fig. 10. This transfer of circulation was also observed in [15]. The number of vortex elements for this simulation remains around 450,000 by the end of the simulation. The linear impulse is preserved within 0.7 %.

Finally, we note that the computational time spent for the interpolation step remains indeed small compared to that spent on the calculation of convection for all the simulations reported in this paper. The difference becomes more pronounced as the number of vortex elements increases. In a numerical experiment using 2 million vortex elements on 384 SP POWER3 processors, the computational time for one interpolation step was less than 10% of the computational time of one single prediction step.

5.2. Nonlinear reaction–diffusion system

In this section, we show that the interpolation kernels can be used for treating nonlinear problems as well. We consider the following one-dimensional reaction–diffusion problem:

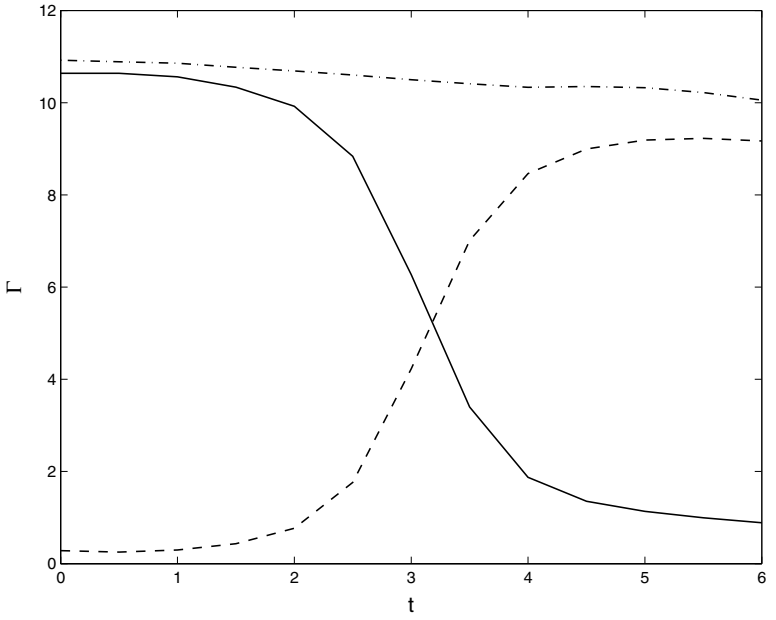

Fig. 10. Evolution of circulation around interacting vortex tubes. Solid, circulation around the cross-section of an inner core on the $z = 0$ plane; dashed, circulation around the cross-section of a bridge on the $x = 0$ plane; dash-dot, the sum of these two circulations.

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial x}\left(\theta^2\frac{\partial\theta}{\partial x}\right) - 2\theta\left(\frac{\partial\theta}{\partial x}\right)^2 \quad \text{for } x \in (0,1),\ t > 0, \tag{39}$$

where the initial and the boundary conditions are given as follows:

$$\theta(x,0) = \theta_{\mathrm{I}}(x) = \frac{2}{3+2x}, \tag{40}$$

$$\frac{\partial\theta}{\partial x} = 0 \quad \text{for } x = 0 \text{ and } 1,\ t > 0. \tag{41}$$

Eq. (39) models thermal conduction in solid crystalline molecular hydrogen, and the following analytical solution was obtained in [28]:

$$\begin{aligned} \theta &= \overline{\theta} + \pi^{-1/2}\int_{\mathbf{R}} A'(\hat{x} + 2t^{1/2}\beta)\exp(-\beta^2)\mathrm{d}\beta, \\ x &= \overline{\theta}\hat{x} + \pi^{-1/2}\int_{\mathbf{R}} A(\hat{x} + 2t^{1/2}\beta)\exp(-\beta^2)\mathrm{d}\beta, \end{aligned} \tag{42}$$

in which $\hat{x}$ is an extensible distance coordinate,

$$\overline{\theta} = \left(\int_0^1 \theta_{\mathrm{I}}(x)^{-1}\mathrm{d}x\right)^{-1} = \frac{1}{2}, \tag{43}$$

and A is odd, of period $2/\overline{\theta}$, and given over a half period by solving

$$\int_0^{A(\gamma)+\overline{\theta}\gamma} \theta_{\mathrm{I}}(x)^{-1}\,\mathrm{d}x = \gamma, \tag{44}$$

which yields

$$A(\gamma) = \begin{cases} \frac{1}{2}\left(-3-\gamma+\sqrt{9+8\gamma}\right) & 0 \leqslant \gamma \leqslant 2, \\ -A(-\gamma) & -2 < \gamma < 0, \\ \text{periodic in 4} & \text{otherwise.} \end{cases} \tag{45}$$

A' is the derivative of A. The expression in (42) can be easily integrated numerically to obtain the pointwise value of the solution with high accuracy.

To solve the problem numerically, we employ the following procedure:

(1) Initialize the particles: set $n = 0$, $x_j^n = (\mathrm{i} - 1/2)\Delta x$, and $\Gamma_j^n = \theta_{\mathrm{I}}(x_j^n)\Delta x$.
(2) Solve for reaction first. The equation $\frac{\partial\theta}{\partial t} = -2\theta(\frac{\partial\theta}{\partial x})^2$ is modeled by a set of ordinary differential equations $\frac{\partial\Gamma_j}{\partial t} = -2\Gamma_j(\underline{\partial_x\theta}_j)^2$ for the particles, where $\underline{\partial_x\theta}_j = \underline{\partial_x\theta}(x_j)$ is the numerical approximation of $(\frac{\partial\theta}{\partial x})_{x=x_j}$. Similarly, we denote the numerical approximation of $\theta(x_j)$ as $\underline{\theta}_j = \underline{\theta}(x_j)$. The ordinary differential equations are discretized in time using the improved polygon method [2].
(3) Solve for diffusion. The equation $\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial x}(\theta^2\frac{\partial\theta}{\partial x})$ is approximated by remeshing using the Λ_3' interpolation kernel (16). $v(x_j)$ and $(\frac{\partial v}{\partial x})_{x=x_j}$ are evaluated from the values of $\underline{\theta}_j$ and $\underline{\partial_x\theta}_j$. in this case, the absence of convection forces the particles to stay at the grid location all the time. Hence, to demonstrate the capability of the method in performing remeshing and diffusion concurrently, the uniform grid for the target particle locations is obtained by shifting the initial particle locations by a distance randomly selected at each time step, i.e., $x_j^n = x_j^0 + \rho^n$, where ρ^n is a random number in $[-\frac{\Delta x}{2}, \frac{\Delta x}{2})$. To satisfy the boundary condition, the particles generated outside the domain during the remeshing process are reflected back into the domain. For example, a particle at $x_j^n < 0$ is moved to $-x_j^n$ at the end of the diffusion substep without changing its strength. In a similar way, we move the particles with $x_j^n > 1$ to $2 - x_j^n$.
(4) Advance time by Δt and repeat steps (2) and (3).

To evaluate $\underline{\theta}(x)$ and $\underline{\partial_x\theta}(x)$, we use the following expressions:

$$\begin{aligned} \underline{\theta}(x) &= \sum_{i=1}^{N} \Gamma_i(\phi_\sigma(x-x_i) + \phi_\sigma(x+x_i) + \phi_\sigma(x-2+x_i)), \\ \underline{\partial_x\theta}(x) &= \frac{\partial\underline{\theta}}{\partial x} = \sum_{i=1}^{N} \Gamma_i\left(\phi_\sigma'(x-x_i) + \phi_\sigma'(x+x_i) + \phi_\sigma'(x-2+x_i)\right). \end{aligned} \tag{46}$$

The two additional terms in the summation represent the image particles included to satisfy the boundary condition. We use a cutoff function $\phi_\sigma \in C_{\mathrm{c}}^4(\mathbf{R})$

$$\phi_\sigma(x) = \frac{1}{\sigma}\phi\left(\frac{x}{\sigma}\right), \tag{47}$$

where

$$\phi(x) = \begin{cases} \frac{693}{512}(1 - 5x^2 + 10x^4 - 10x^6 + 5x^8 - x^{10}) & |x| < 1, \\ 0 & 1 \leqslant |x|. \end{cases} \tag{48}$$

For $\operatorname{supp}\phi_\sigma = [-\sigma, \sigma]$, the image particles included in (46) are enough to satisfy the correct boundary conditions for $\sigma < 1$. Other cutoff functions may be used as long as they are of compact support and in $C_B^3(\mathbf{R})$, which is required to obtain convergence in L^∞ as shown in Appendix A.

Computations were performed for $t = [0, 0.011]$ in 4-byte precision on a Pentium 4 workstation. We chose $\Delta t/\Delta x^2 = 1.1$ to satisfy the non-negativity constraint (29). Fig. 11 and Table 1 show the convergence of the approximate solution to the analytical solution with the numerical parameters being refined. Note that both σ and $\Delta x/\sigma$ should approach zero to suppress noise at high wave numbers. One can also notice that the numerical error is more prominent near the boundary. The initial conditions with their images are only continuous at the boundary, and hence the error near the boundary in the initial discretization is larger than that in the domain interior. However, the overall trend of convergence is clear.

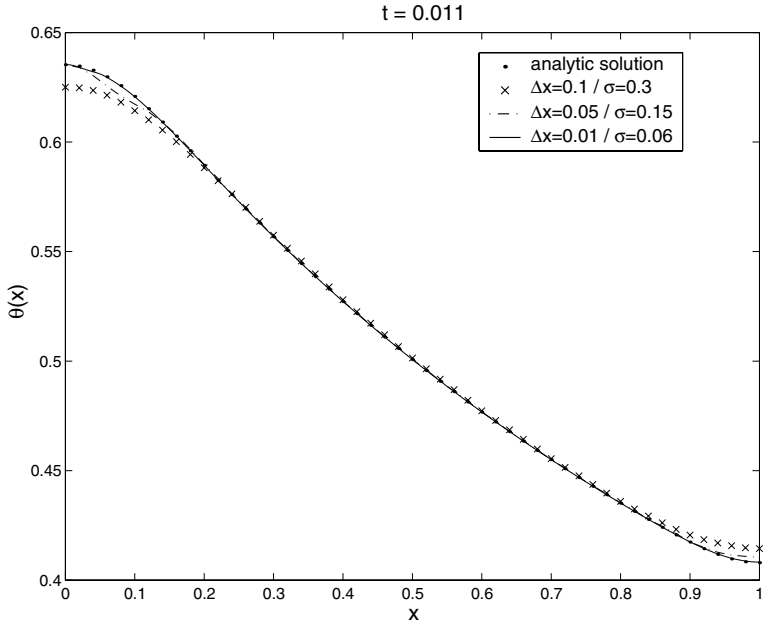

Fig. 11. Profiles at $t = 0.011$ for different numerical parameters compared against the analytical solution obtained from (42).

Table 1
L^∞ error for different numerical parameters at $t = 0.011$

	$\sigma/\Delta x = 3$	$\sigma/\Delta x = 6$
$\Delta x = 0.1$	1.0266×10^{-2}	
$\Delta x = 0.05$	4.7675×10^{-3}	
$\Delta x = 0.01$	8.7498×10^{-3}	8.6109×10^{-4}
$\Delta x = 0.005$		4.6580×10^{-4}

6. Summary

In this article, we propose a scheme to treat diffusion and remeshing, simultaneously, in Lagrangian vortex methods. Interpolation kernels similar to those that have been used for remeshing of particle distributions in inviscid vortex simulations are obtained by utilizing the moment-based redistribution method. The stability properties of the new interpolation kernels were investigated by using analogies to Eulerian schemes. Numerical examples show that the scheme works well in test problems. Results suggest that the scheme can be successfully applied to complex problems, including cases in which nonlinear diffusion plays an important role.

Our work suggests a number of avenues for future research. For instance, the characterization of a particle distribution through its moments suggests the possibility of an adaptive method for redistribution. Not only the δ distribution but also the cutoff function can be explicitly considered during the redistribution process [21,29,33]. By characterizing the cutoff function of a particle through its moments, one may develop redistribution formulae for particles with different cutoff radii, endowing Lagrangian vortex methods with multi-level capabilities.

Acknowledgements

The research was supported by the Mathematical, Information, and Computational Sciences (MICS) program of the Office of Science in the US Department of Energy under the grant number DE-FG02-98ER25355.

Computational support for large-scale scientific simulations was provided by the National Energy Research Scientific Computing Center (NERSC). The authors wish to thank the reviewers for their helpful comments, which contributed to improving the manuscript.

Appendix A. Consistency of the redistribution scheme (9)

In [33], the convergence of the redistribution method was established for the case of constant diffusivity. The argument used in that work relied heavily on the fact that the fundamental solution in this case is explicitly given by a Gaussian distribution. Since the fundamental solution of (1) is not a Gaussian distribution, the argument given in [33] is not directly applicable to the analysis of the redistribution formulae (9). In this section, we provide a brief proof for the consistency of these formulae. Once the consistency is established, the convergence follows by having an additional condition on stability.

To establish consistency, we use the following estimate for the fundamental solution of a uniformly parabolic linear equation [20]:

$$|D_t^r D_x^s Z(x,\xi,t,\tau)| \leqslant c(t-\tau)^{-\frac{d+2r+s}{2}} \exp\left(-C\frac{|x-\xi|^2}{t-\tau}\right), \tag{A.1}$$

where $2r+s \leqslant 2$. Two consequences of this estimate are exploited in the following discussion. The first is quite straightforward. Let f be a function, which is continuous and globally Lipschitz with a constant K in $\mathbf{R}^d$. Then, we can show that

$$\left|\int_{\mathbf{R}^d} Z(x,\xi,t,t-\Delta t)f(x)\mathrm{d}x - f(\xi)\right| \leqslant cK\sqrt{\Delta t}\int_{\mathbf{R}^d}|z|\exp(-Cz^2)\mathrm{d}z, \tag{A.2}$$

which in turn shows that $|\int_{\mathbf{R}^d} Z(x,\xi,t,t-\Delta t)f(x)\mathrm{d}x - f(\xi)| = \mathrm{O}(\sqrt{\Delta t})$. The other important consequence is that

$$\begin{aligned}\int_{|x-\xi|\geqslant\left(\sqrt{\Delta t}\right)^{1-\epsilon}} |Z(x,\xi,t,t-\Delta t)||x-\xi|^m\mathrm{d}x &\leqslant \int_{|x-\xi|\geqslant\left(\sqrt{\Delta t}\right)^{1-\epsilon}} c\Delta t^{-d/2}\exp\left(-C\frac{|x-\xi|^2}{\Delta t}\right)|x-\xi|^m\mathrm{d}x \\ &\leqslant \int_{|z|\geqslant\left(\sqrt{\Delta t}\right)^{-\epsilon}} c\exp(-Cz^2)\left|z\sqrt{\Delta t}\right|^m\mathrm{d}z = \mathrm{o}(\Delta t^q)\end{aligned} \tag{A.3}$$

for $0<\epsilon<1$, $q>0$ and $m\geqslant 0$.

With these estimates at hand, we may show the consistency of the redistribution formulae (9) in the sense that the new particle distribution truly approximates the fundamental solution. Each source particle is assumed to be redistributed into target particles within a ball of radius R, centered at the location of the source particle. we assume that R scales as $\mathrm{O}(h)$, where $h=\sqrt{\Delta t}$.

First, we show that each moment evolves consistently. For the 0th moment, $G_{0,i}^n = \underline{G}_{0,i}^n = 1$, if f_{ij}^n satisfies (9). Thus the moment matching condition is satisfied exactly. For $G_{1,i}^n$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(G_{1,i}^n - \underline{G}_{1,i}^n\right) = \int_{\mathbf{R}} Z(x,x_i^{n-1},t,0)\frac{\mathrm{d}v}{\mathrm{d}x}\mathrm{d}x - \left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)_{x=x_i^{n-1}} = \mathrm{O}(\Delta t^{1/2})$$

by the application of (A.2). Thus, $|\ G_{1,i}^n - \underline{G}_{1,i}^n\ | = \mathrm{O}(\Delta t^{3/2})$. For $G_{2,i}^n$, we separate the integral into two parts.

$$\left|\frac{\mathrm{d}}{\mathrm{d}t}\left(G_{2,i}^n - \underline{G}_{2,i}^n\right)\right| \leqslant \int_{\mathbf{R}}\left|2v(x)-2v(x_i^{n-1})\right|Z(x,x_i^{n-1},t,0)\mathrm{d}x + \int_{\mathbf{R}}\left|2(x-x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right|Z(x,x_i^{n-1},t,0)\mathrm{d}x.$$

The first term is $\mathrm{O}(\Delta t^{1/2})$ by the application of (A.2). The second term is estimated as follows:

$$\begin{aligned}\int_{\mathbf{R}}\left|2(x-x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right|Z(x,x_i^{n-1},t,0)\mathrm{d}x &\leqslant \int_{\Omega}\left|2(x-x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right|Z(x,x_i^{n-1},t,0)\mathrm{d}x \\ &\quad + \int_{\Omega^c}\left|2(x-x_i^{n-1})\frac{\mathrm{d}v}{\mathrm{d}x}\right|Z(x,x_i^{n-1},t,0)\mathrm{d}x,\end{aligned}$$

where $\Omega = \{x \in \mathbf{R},\ \text{s.t. } |x - x_i^{n-1}| < R'h\}$, where $R' = \mathrm{O}(h^{-\epsilon})$. It can be shown that the first term is $\mathrm{O}(R'h) = \mathrm{O}(\Delta t^{1/2-\epsilon})$, and the second term is $\mathrm{o}(h^q)$ for all $q > 0$ from (A.3). Thus, we have $|G_{2,i}^n - \underline{G}_{2,i}^n| = \mathrm{O}(\Delta t^{3/2-\epsilon})$.

Therefore, for the ith source particle, the redistribution formulae (9) give f_{ij}^n satisfying the following conditions around x_i^{n-1} for $M = 1$, i.e.,

$$\forall |k| \leqslant M+1, \quad \exists C_{k,M'},\ \text{s.t. } \forall i,n, \left|\underline{G}_{k,i}^n - G_{k,i}^n\right| \leqslant C_{k,M'} h^{M'} \Delta t \tag{A.4}$$

for any M' satisfying $M - 1 < M' < M$.

We are finally at the stage where we can estimate the error between the new particle distribution and the fundamental solution using (A.4). From the Taylor series remainder theorem of a test function ϕ, we get

$$\begin{aligned}
&\forall \phi \in C_B^{M+2}(\mathbf{R}),\\
&\left|\int_{\mathbf{R}} \left(\sum_{j=1}^{N} f_{ij}^n \delta(x - x_j^n) - Z(x, x_i^{n-1}, \Delta t, 0)\right)\phi(x)\mathrm{d}x\right|\\
&\quad \leqslant \sum_{|k|\leqslant M+1} \frac{1}{k!}\|\phi\|_{C_B^{M+2}(\mathbf{R})} \left|\int_{\mathbf{R}} (x - x_i^{n-1})^k \left(\sum_{j=1}^{N} f_{ij}^n \delta(x - x_j^n) - Z(x, x_i^{n-1}, \Delta t, 0)\right)\mathrm{d}x\right|\\
&\qquad + \sum_{|k|=M+2} \frac{1}{k!}\|\phi\|_{C_B^{M+2}(\mathbf{R})} \left|\int_{\Omega} (x - x_i^{n-1})^k \left(\sum_{j=1}^{N} f_{ij}^n \delta(x - x_j^n) - Z(x, x_i^{n-1}, \Delta t, 0)\right)\mathrm{d}x\right|\\
&\qquad + \sum_{|k|=M+2} \frac{1}{k!}\|\phi\|_{C_B^{M+2}(\mathbf{R})} \left|\int_{\Omega^c} (x - x_i^{n-1})^k Z(x, x_i^{n-1}, \Delta t, 0)\mathrm{d}x\right|.
\end{aligned} \tag{A.5}$$

The first term in the right hand side is $\mathrm{O}(h^{M'}\Delta t)$ by the assumed moment conditions, and the last term is $\mathrm{o}(h^q)$ for all $q > 0$ by the estimate (A.3). The second term is $\mathrm{O}(R'^{M+2}h^M\Delta t)$, because

$$\forall x \in \Omega, \quad \left|(x - x_i^{n-1})^{(k)}\right| \leqslant (R'h)^{M+2} = R'^{M+2}h^M\Delta t.$$

Since $R = \mathrm{O}(h)$, we may take ϵ as small as we want. Therefore, we have the following result for all $M' < M$:

$$\left\|\sum_{j=1}^{N} f_{ij}^n \delta(x - x_j^n) - Z(x, x_i^{n-1}, \Delta t, 0)\right\|_{(C_B^{M+2}(\mathbf{R}))'} \leqslant C h^{M'} \Delta t, \tag{A.6}$$

Using a standard argument with the additional condition of stability, we can also show that the global truncation error behaves as $\mathrm{O}(h^{M'})$, where $M' < M$. This is not as sharp as what we had for the case of constant diffusivity considered in [33], where the global truncation error was $\mathrm{O}(h^M)$.

We have shown consistency in the distribution sense [12,37], i.e., in $(C_B^{M+2}(\mathbf{R}))'$, not in the typical L^p spaces. The reason is that the spirit of the method can be more clearly recognized in terms of the approximation of the fundamental solution by δ distributions, which cannot be treated in L^p. However, it is easy to show that the methods also generate a convergent sequence in L^p by convolving the particle distribution with a sufficiently regular cutoff function ϕ. For example, given $\phi \in C_B^{M+2}(\mathbf{R}^d)$, we define $\phi_\sigma(x) \equiv \frac{1}{\sigma^d}\phi(\frac{x}{\sigma})$. Then, for a regular enough solution u,

$$\|u - \underline{u} * \phi_\sigma\|_\infty \leqslant \|u - u * \phi_\sigma\|_\infty + \|(u - \underline{u}) * \phi_\sigma\|_\infty,$$

where $\underline{u}$ is the approximation made up with a linear superposition of δ distributions, i.e., $\|u - \underline{u}\|_{(C_B^{M+2}(\mathbf{R}))'} = \mathrm{O}(h^{M'})$. It is easy to see that

$$\begin{aligned}
&\exists m_1, m_2 > 0,\ \text{s.t. } \|u - u * \phi_\sigma\|_\infty = \mathrm{O}(\sigma^{m_1}),\\
&\|(u - \underline{u}) * \phi_\sigma\|_\infty \leqslant \sup_{x\in\mathbf{R}^d} \left|\int_{\mathbf{R}^d} (u - \underline{u})(y)\phi_\sigma(x - y)\mathrm{d}y\right| \leqslant \|(u - \underline{u})\|_{\left(C_B^{M+2}(\mathbf{R})\right)'} \|\phi_\sigma\|_{C_B^{M+2}(\mathbf{R})} = \mathrm{O}(h^{M'}/\sigma^{m_2}).
\end{aligned}$$

This shows that the error in L^∞ can be estimated as $\mathrm{O}(\sigma^{m_1}) + \mathrm{O}(h^{M'}/\sigma^{m_2})$. We note that this error estimate does not include any detailed consideration on the contribution of the error from initial discretization, and

the error scaling as $O(\sigma^{m_1})$ is due to variable diffusivity, and not due to initial discretization. This additional error does not occur when one only deals with the case of constant diffusivity, where one can use convolution to separate the cutoff function from the error estimate, assuming that the cutoff function behaves well. However, in the case of variable diffusivity, the independency of the error estimate from the cutoff function is lost even in the redistribution method.

References

[1] L.A. Barba, A. Leonard, C.B. Allen, Advances in viscous vortex methods—meshless spatial adaption based on radial basis function interpolation, International Journal for Numerical Methods in Fluids 47 (2005) 387–421.
[2] S.C. Chapra, R.P. Canale, Numerical Methods for Engineers, second ed., McGraw-Hill, 1988.
[3] P. Chatelain, A. Leonard, Face-centred cubic lattices and particle redistribution in vortex methods, Journal of Turbulence 3 (46) (2002) 1–6, doi:10.1088/1468-5248/3/1/046.
[4] R.C.Y. Chin, G.W. Hedstrom, A dispersive analysis for difference schemes: tables of generalized Airy functions, Mathematics of Computation 32 (1978) 1163–1170.
[5] A.J. Chorin, Numerical study of slightly viscous flow, Journal of Fluid Mechanics 57 (1973) 785–796.
[6] G.-H. Cottet, P.D. Koumoutsakos, Vortex Methods: Theory and Practice, Cambridge University Press, 2000.
[7] G.-H. Cottet, P.D. Koumoutsakos, M.L. Ould-Salihi, Vortex methods with spatially varying cores, Journal of Computational Physics 162 (2000) 164–185.
[8] G.-H. Cottet, S. Mas-Gallic, A particle method to solve the Navier–Stokes system, Numerische Mathematik 57 (1990) 805–827.
[9] P. Degond, S. Mas-Gallic, The weighted particle method for convection–diffusion equations, Part 1: The case of an isotropic diffusion, Mathematics of Computation 53 (1989) 485–507.
[10] P. Degond, S. Mas-Gallic, The weighted particle method for convection–diffusion equations, Part 2: The anisotropic case, Mathematics of Computation 53 (1989) 509–525.
[11] P. Degond, F.-J. Mustieles, A deterministic approximation of diffusion-equations using particles, SIAM Journal on Scientific and Statistical Computing 11 (1990) 293–310.
[12] G. Friedlander, M. Joshi, Introduction to the Theory of Distributions, second ed., Cambridge University Press, 1998.
[13] A. Gharakhani, A higher order vorticity redistribution method for 3-D diffusion in free space, Sandia Report, SAND2000-2505, 2000.
[14] A.F. Ghoniem, G. Heidarinejad, A. Krishnan, Numerical simulation of a thermally stratified shear layer using the vortex element method, Journal of Computational Physics 79 (1988) 135–166.
[15] S. Kida, M. Takaoka, F. Hussain, Collision of two vortex rings, Journal of Fluid Mechanics 230 (1991) 583–646.
[16] O.M. Knio, A.F. Ghoniem, Numerical study of a three-dimensional vortex method, Journal of Computational Physics 86 (1990) 75–106.
[17] O.M. Knio, A.F. Ghoniem, Three-dimensional vortex simulation of rollup and entrainment in a shear layer, Journal of Computational Physics 97 (1991) 172–223.
[18] P.D. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, Journal of Computational Physics 138 (1997) 821–857.
[19] P.D. Koumoutsakos, A. Leonard, F. Pepin, Boundary conditions for viscous vortex methods, Journal of Computational Physics 113 (1994) 52–56.
[20] O.A. Ladyženskaja, V.A. Solonnikov, N.N. Ural'ceva, Linear and Quasi-linear Equations of Parabolic Type, first ed., American Mathematical Society, 1968.
[21] I. Lakkis, A.F. Ghoniem, Axisymmetric vortex method for low Mach number, diffusion-controlled combustion, Journal of Computational Physics 184 (2003) 435–475.
[22] R.J. LeVeque, Numerical Methods for Conservation Laws, second ed., Birkhäuser Verlag, 1992.
[23] K. Lindsay, R. Krasny, A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow, Journal of Computational Physics 172 (2001) 879–907.
[24] Z.Y. Lu, T.J. Ross, Diffusing-vortex numerical scheme for solving incompressible Navier–Stokes equations, Journal of Computational Physics 95 (1991) 400–435.
[25] A.J. Majda, A.L. Bertozzi, Vorticity and Incompressible Flow, first ed., Cambridge University Press, 2002.
[26] Y.M. Marzouk, A.F. Ghoniem, *k*-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical *N*-body simulations, Journal of Computational Physics 207 (2005) 493–528.
[27] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, M.S. Warren, 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 178 (2002) 427–463.
[28] G. Rosen, Nonlinear heat conduction in solid H_2, Physical Review B 19 (1979) 2398–2399.
[29] L.F. Rossi, Resurrecting core spreading vortex methods: a new scheme that is both deterministic and convergent, SIAM Journal of Scientific Computing 17 (1996) 370–397.
[30] N. Rott, B.J. Cantwell, Vortex drift. I: Dynamic interpretation, Physics of Fluids A 5 (1993) 1443–1450.
[31] G. Russo, A deterministic vortex method for the Navier–Stokes equations, Journal of Computational Physics 108 (1992) 84–94.
[32] S. Shankar, Grid-free redistribution methods for axisymmetric and anisotropic diffusion, Internal Report, Reacting Gas Dynamics Laboratory, Massachusetts Institute of Technology, 1999.

[33] S. Shankar, L. van Dommelen, A new diffusion procedure for vortex methods, Journal of Computational Physics 127 (1996) 88–109.
[34] K. Shariff, A. Leonard, Vortex rings, Annual Review of Fluid Mechanics 24 (1992) 235–279.
[35] S.K. Stanaway, B.J. Cantwell, P.R. Spalart, Navier–Stokes simulations of axisymmetric vortex rings, AIAA Technical Paper, AIAA-88-0318, 1988.
[36] S.K. Stanaway, B.J. Cantwell, P.R. Spalart, A numerical study of viscous vortex rings using a spectral method, NASA Technical Memorandum, NASA TM 101041, 1988.
[37] R.S. Strichartz, A Guide to Distribution Theory and Fourier Transforms, first ed., World Scientific Publishing, 1994.
[38] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, first ed., Wadsworth & Brooks/Cole, 1989.
[39] F. Thirifay, G.S. Winckelmans, Development of a Lagrangian method for combustion and application to the planar methane air jet diffusion flame, Journal of Turbulence 3 (59) (2002) 1–7, doi:10.1088/1468-5248/3/1/059.
[40] H. Trac, U. Pen, A primer on Eulerian computational fluid dynamics for astrophysics, Publications of the Astronomical Society of the Pacific 115 (2003) 303–321.
[41] G.S. Winckelmans, A. Leonard, Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows, Journal of Computational Physics 109 (1993) 247–273.
[42] G.S. Winckelmans, J.K. Salmon, M.S. Warren, A. Leonard, Application of fast parallel and sequential tree codes to computing three-dimensional flows with the vortex element and boundary element methods, ESAIM Proceedings: Vortex Flows and Related Numerical Methods II 1 (1996) 225–240.