MULTISCALE MODEL. SIMUL.
Vol. 4, No. 1, pp. 328–357


# MULTILEVEL ADAPTIVE PARTICLE METHODS FOR CONVECTION-DIFFUSION EQUATIONS*

MICHAEL BERGDORF†, GEORGES-HENRI COTTET‡, AND PETROS KOUMOUTSAKOS†

**Abstract.** We present novel multilevel particle methods with extended adaptivity in areas where increased resolution is required. We present two complementary approaches as inspired by r-adaptivity and adaptive mesh refinement (AMR) concepts introduced in finite difference and finite element schemes. For the r-adaptivity a new class of particle-based mapping functions is introduced, while the particle AMR method uses particle remeshing in overlapping domains as a key element. The advantages and drawbacks of the proposed particle methods are illustrated based on results from the solution of one-dimensional convection-diffusion equations, while the extension of the method to higher dimensions is demonstrated in simulations of the inviscid evolution of an elliptical vortex.

**Key words.** particle methods, adaptive mesh refinement, r-adaptivity, convection-diffusion equations, vortex methods

**AMS subject classifications.** 76M28, 65M55, 65M50

**DOI.** 10.1137/040602882

## 1. Introduction.

Particle methods, when applied to the solution of convection-diffusion equations in the context of vortex methods and smooth particle hydrodynamics, enjoy an automatic adaptivity of the computational domain as dictated by the flow map. The field quantities can always be reconstructed by a linear superposition of the individual fields carried by the particles. In smooth particle methods—as opposed to point particle methods—each particle is associated with a smooth core function, or "blob", enabling the smooth representation of the field quantities and efficient discretizations of the governing equations.

The Lagrangian form of particle methods avoids the explicit discretization of the convective term in the governing transport equations and the associated stability constraints. The particle positions are modified according to the local flow map, making the method self-adaptive. This adaptation comes at the expense of the regularity of the particle distribution as particles move in order to adapt to the gradients of the flow field. Particle regularity is best guaranteed by remeshing the particle locations on a regular grid. An important particular case where remeshing and self-adaptivity can be enforced together is when the topology of the solution can be tracked and accounted for during the computation. This is the case for inviscid flow calculations of vortex sheets (see, for instance, [13, 14]). In those examples remeshing can be done along filaments or sheets of vorticity, which guarantees the accuracy of the computations for a minimal cost, as particles remain located in the vorticity support.

For more general cases, regularity in the particle resolution in general imposes a severe restriction on the overall adaptivity of particle methods. For example, in bluff body flows the boundary of the body is the source of vorticity in the flow, and it is important to discretize adequately the boundary layer region near the surface of the body. This requirement dictates the size of the particle cores. However, for

*Received by the editors January 5, 2004; accepted for publication (in revised form) August 4, 2004; published electronically June 22, 2005.
http://www.siam.org/journals/mms/4-1/60288.html
†Institute of Computational Science, ETH Zürich, 8092 Zürich, Switzerland (bergdorf@inf.ethz.ch, petros@inf.ethz.ch).
‡LMC-IMAG, Université Joseph Fourier, Grenoble, France (georges-henri.cottet@imag.fr.).

constant-size particles, as the vorticity gradients decay in the wake, it is clear that the flow is discretized using unnecessarily large numbers of computational elements. This deficiency of constant-size particle methods clearly detracts from the adaptive character of the method and its capability to accurately resolve vorticity gradients while remaining computationally efficient.

Hence, beyond adaptivity as dictated by the flow map it is often necessary to employ particle methods with different resolution requirements as dictated by the physics of the flow. In this case particle methods with variable-size particles have been developed in the context of vortex methods for incompressible flows.

In [10] Hou proved the convergence of vortex methods with variable core sizes for the Euler equations. In [17] this proof was extended to the viscous case, and the method was used for the simulation of wakes with stretched particle resolution. In [6] Cottet, Koumoutsakos, and Ould–Salihi formulated a convergent variable core size vortex method for the Navier–Stokes equations by using mappings from a reference space with uniform blobs to the "physical" space with blobs of varying size in conjunction with an anisotropic diffusion operator. This method was extended in a domain decomposition framework to handle several mappings corresponding to different zones and grid-size requirements in the flow.

All these methods require prior knowledge of where the flow field should be refined, and the refinement strategy is not of adaptive nature. In this paper we propose and validate for one-dimensional (1D) model problems two classes of techniques that allow us to refine dynamically the computational domain and adapt accordingly its particle discretization. Both types of methods can be viewed as extensions of the methods proposed in [6]: one uses a global adaptive mapping, while the other is based on combinations of several local mappings. They also relate to adaptive Eulerian methods. The first class pertains to r-adaptive finite element methods, while the second one is reminiscent of finite difference adaptive mesh refinement (AMR) methods.

The concept of r-adaptivity originated in the realm of finite elements and amounts to moving the computational elements into areas of the computational domain where increased resolution capability is needed. The first finite element method that achieved improved accuracy by adaptively moving the nodes of the triangulation were presented by Miller and Miller in [15]. The equations of motion of the nodes have been determined by minimizing a global error functional with respect to the weights of the finite element basis functions and the positions of the nodes simultaneously. Since then, many variations of this concept have been produced; for instance, in [11] the minimization of the error functional has been replaced by so-called moving mesh partial differential equations which can be derived from node equidistribution principles.

One can argue that particle methods are inherently r-adaptive due to their Lagrangian character. This is true in the sense that computational elements are moved into areas where increased capability of resolution is required. Unlike the finite element method, the characteristic length scale of these elements remains unaltered and usually uniform. Thus, the analogue for r-adaptivity in particle methods is to employ particles with varying core sizes which adapt with the evolution of the solution they represent.

AMR methods were first proposed by Berger and Oliger [1] in the context of finite difference methods. The idea here is to define blocks of uniform grid sizes that are defined dynamically based, for instance, on a posteriori error estimates. Blocks with different grid sizes communicate by exchange of boundary conditions, as in domain

decomposition methods. We revisit this class of methods in the context of particle methods by merely extending the variable-blob techniques introduced in [6] for the case of several local mappings. Our method is heavily based on overlapping of the subdomains, and particle remeshing in the overlapping zones plays the role of interface boundary conditions in finite difference AMR methods.

In our two classes of techniques, one can expect that rephrasing in terms of particle method concepts inherited from adaptive finite element or finite difference methods may lead to methods that will keep the essence of particle methods, namely, its robustness when dealing with convection dominated problems or problems involving complex physics, while optimizing their accuracy.

An outline of the paper is as follows. In section 2 we first briefly recall the framework of particle methods with uniform or spatially varying cores. We then develop in section 3 particle methods with a global adaptive mapping. In section 4 we turn to the AMR particle method, based on several local mappings. Section 5 is devoted to numerical illustrations for the 1D viscous Burgers equation and preliminary results for the two-dimensional (2D) incompressible Euler equations. Finally, in section 6 we draw some conclusions and discuss potential for improvement in the multidimensional case.

**2. Particle methods for convection-diffusion equations.** Particle methods are suitable for solving conservation laws such as

$$\frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{u}q) = \nu \Delta q. \tag{2.1}$$

They discretize the Lagrangian form of (2.1) expressed as

$$\begin{aligned} \frac{dq}{dt} &= \nu \Delta q, \\ \frac{d\mathbf{x}}{dt} &= \mathbf{u}, \\ \frac{dv}{dt} &= \nabla \cdot \mathbf{u}. \end{aligned} \tag{2.2}$$

Particle methods rely on approximating the function $q$ using a linear superposition of particle quantities so that

$$q(\mathbf{x}, t) \approx \sum_j Q_j(t) \zeta^\epsilon(\mathbf{x} - \mathbf{x}_j(t)), \tag{2.3}$$

where $Q_j$ are the particle strengths which are usually selected as $Q_j = v_j q(\mathbf{x}_j)$, $v_j$ being the particle volumes and $\zeta^\epsilon(\mathbf{x}) \equiv \frac{1}{\epsilon^d}\zeta(\mathbf{x}/\epsilon)$ being a mollification of the Dirac–Delta function. Differential operators such as the Laplacian are replaced by suitable integral operators which are discretized using the particle positions as quadrature points. For example, the method of particle strength exchange (PSE) [8] approximates the Laplacian as

$$\Delta q \approx \Delta^\epsilon q \equiv \epsilon^{-2} \int_\Omega [q(\mathbf{y}) - q(\mathbf{x})] \eta^\epsilon(|\mathbf{x} - \mathbf{y}|)\, d\mathbf{y}, \tag{2.4}$$

where the radially symmetric kernel $\eta^\epsilon(x) = \frac{1}{\epsilon^d}\eta(x/\epsilon)$ has to satisfy

$$\int_\Omega x_i^2 \eta \, d\mathbf{x} = 2, \quad i = 1, \ldots, d. \tag{2.5}$$

By using the above operator approximation (2.2) is solved by integrating the following set of ODEs for the particle positions, volumes, and strengths:

$$\begin{aligned} \frac{d\mathbf{x}_j}{dt} &= \mathbf{u}(\mathbf{x}_j, t), \\ \frac{dv_j}{dt} &= \nabla \cdot \mathbf{u}(\mathbf{x}_j, t) v_j, \\ \frac{dQ_j}{dt} &= \frac{\nu}{\epsilon^2} \sum_k [Q_k v_j - Q_j v_k] \eta^\epsilon(|\mathbf{x}_j - \mathbf{x}_k|) \end{aligned} \tag{2.6}$$

for $j = 1, \ldots, N$.

**2.1. Particle method with variable core sizes.** In order to resolve problems with variable resolution requirements, particle methods with variable core sizes have been introduced in [10] and [6]. They introduce a smooth map $f : \hat{\Omega} \to \Omega$, which maps a "reference" space $\hat{\Omega} \subseteq \mathbb{R}^d$ with uniform core sizes $\hat{\epsilon}$ onto a "physical" space $\Omega \subseteq \mathbb{R}^d$, so that

$$\mathbf{x} = \mathbf{f}(\hat{\mathbf{x}}), \quad \hat{\mathbf{x}} = \mathbf{g}(\mathbf{x}), \quad \{\underline{\Phi}\}_{ij} = \frac{\partial \hat{x}_i}{\partial x_j}, \quad \text{and} \quad |\underline{\mathbf{\Phi}}| = \det \underline{\mathbf{\Phi}}, \tag{2.7}$$

and the cores in physical space become anisotropic and of the order of

$$\epsilon \sim \left( \frac{1}{|\underline{\mathbf{\Phi}}|} \right)^{\frac{1}{d}}. \tag{2.8}$$

As shown in [6] the Laplacian can be written in reference space as

$$\Delta q = |\underline{\mathbf{\Phi}}| \frac{\partial}{\partial \hat{x}_i} \left( b_{ij} \frac{\partial q}{\partial \hat{x}_j} \right), \tag{2.9}$$

with

$$b_{ij} = \frac{1}{|\underline{\mathbf{\Phi}}|} \frac{\partial \hat{x}_i}{\partial x_k} \frac{\partial \hat{x}_j}{\partial x_k}. \tag{2.10}$$

The operator in reference space is generally of anisotropic nature. An approximation of the kind of (2.4) for the anisotropic case has been developed in [9], and we can approximate (2.9) as

$$\Delta q \approx \frac{|\underline{\mathbf{\Phi}}|}{\hat{\epsilon}^2} \int \psi_{ij}^\epsilon(\hat{\mathbf{x}} - \hat{\mathbf{y}}) \left( \frac{m_{ij}(\hat{\mathbf{x}}) + m_{ij}(\hat{\mathbf{y}})}{2} \right) [q(\hat{\mathbf{y}}) - q(\hat{\mathbf{x}})] \, d\hat{\mathbf{y}}, \tag{2.11}$$

with

$$m_{ij} = b_{ij} - \frac{1}{d+2} \delta_{ij} \delta_{kl} b_{kl}, \tag{2.12}$$

and

$$\psi_{ij}(\mathbf{x}) = x_i x_j \rho(\mathbf{x}), \tag{2.13}$$

with a spherically symmetric $\rho$ which has to obey

$$\int_\Omega x_i^4 \rho(\mathbf{x}) \, d\mathbf{x} = d + 2. \tag{2.14}$$

As in the uniform core size method (2.6), we convect the particles in physical space, but diffusion is performed in reference space, so that with $N$ particles, located in $\{\mathbf{x}_j(t)\}_{j=1}^N = \{\mathbf{f}(\hat{\mathbf{x}}_j)\}_{j=1}^N$ we find an approximate solution to (2.1) by integrating the following set of ODEs:

$$\begin{aligned}
\frac{d\mathbf{x}_j}{dt} &= \mathbf{u}(\mathbf{x}_j, t),\\
\frac{dQ_j}{dt} &= \frac{\nu}{\hat{\epsilon}^2}\sum_k \psi_{pq}^{\hat{\epsilon}}(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_k)\left(\frac{m_{pq}(\hat{\mathbf{x}}_j) + m_{pq}(\hat{\mathbf{x}}_k)}{2}\right)[\hat{v}_j\hat{Q}_k - \hat{v}_k\hat{Q}_j],\\
\frac{d\hat{v}_j}{dt} &= \hat{\nabla}\cdot(\underline{\mathbf{\Phi}}\mathbf{u})(\mathbf{x}_j, t)\hat{v}_j.
\end{aligned} \tag{2.15}$$

In the above equation $Q_j$ and $\hat{Q}_j$ denote the particle strength in physical and reference space, respectively, related by

$$\hat{Q}_j = Q_j|\underline{\mathbf{\Phi}}|(\mathbf{x}_j).$$

**2.2. Time stepping and particle redistribution.** The differential systems (2.6) and (2.15) need to be numerically integrated by a time-stepping algorithm. In practice, methods that are at least second order accurate are used. In the numerical illustrations in section 5 we use a second order midpoint Runge–Kutta algorithm.

Completed by these discretizations, it is not difficult to realize that particle methods are unconditionally linearly stable: for constant velocity fields, particle methods yield exact solutions. Nonlinear stability imposes that particle trajectories do not cross, which results in a time-step constraint of the type

$$\Delta t \le C\|\nabla\mathbf{u}\|_\infty^{-1}, \tag{2.16}$$

where the coefficient $C$ depends on the particular numerical scheme.

A critical issue for the accuracy of particle methods is the regularity of the particle distribution. Large velocity gradients lead to local accumulations or lack of particles that result in a deterioration of the theoretical convergence rates. To maintain the regularity of the particle distribution it is necessary to remesh them on regular locations. This redistribution is in general done through interpolation of particle values. The interpolation function has to be smooth without introducing dissipation. For our applications we use the $M_4'$ function which has been proposed by Monaghan in [16] and validated in [12]. It is a continuously differentiable function that has the property of conserving the first three moments of the particle distribution. Multidimensional remeshing is performed by tensor products of this function. It has been successfully used in a number of complex flow simulations. In variable-size particle methods as defined in section 2.1, remeshing is performed in reference coordinates.

The time scale on which remeshing is done is mostly empirical. It is clearly correlated to the strain rate of the flow, and since the time step itself is constrained by (2.16), a simple rule that proved to be efficient in all our calculations is to remesh at the end of every time step.

Note that remeshing also has some nice consequences on the diffusion approximations. Since relative particle distances do not differ significantly from those of grid points on regular mesh, one may replace the continuous moment properties (2.5) and (2.14) by discrete conditions on localized stencils (for instance three points in one direction). This has the advantage of bypassing the quadrature approximation, and the resulting overlapping condition, involved in the diffusion approximation in its original form.

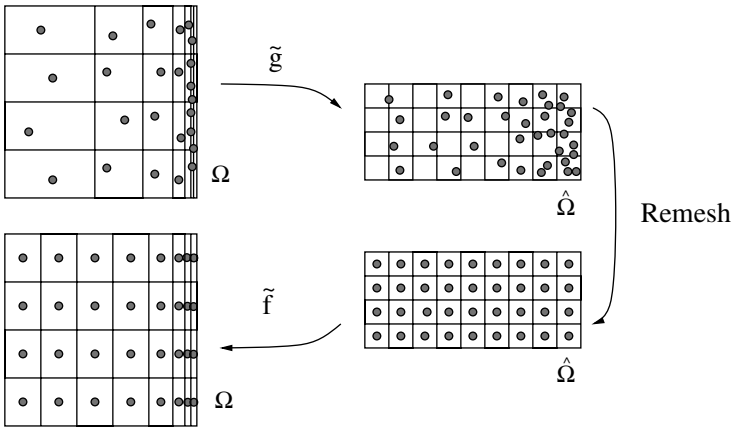


FIG. 1. *Illustration of an invertible adaptive mappings algorithm.*

**3. Particle method with adaptive global mapping.** The framework introduced in section 2.1 involves analytical maps for which the Jacobian is readily available. If the mapping is invertible, it can be changed at any time during a simulation. The particles are convected in physical space, and remeshing and diffusion are performed in reference space, so that one can envision the following adaptive algorithm (see Figure 1): given a map $\mathbf{f}$ and an inverse map $\mathbf{g}$ and particles in $\Omega$,

1. transfer particles to $\hat{\Omega}$ using an *adapted* inverse map $\tilde{\mathbf{g}}$;
2. remesh particles in $\hat{\Omega}$ and perform diffusion;
3. transfer particles to $\Omega$ with *adapted* map $\tilde{\mathbf{f}}$.

Like that, analytic, invertible maps could be dynamically adapted to the flow field. For instance, this could be done by adjusting their parameters: if $f(\hat{x}) = c_1 e^{-c_2 \hat{x}}$ was used as a map to resolve a 1D boundary layer, we could adjust $c_1$ and $c_2$ to account for the growth of the layer. However, in order to have a map that is general enough to provide heterogeneous flexible adaptation, it is desirable to use a finite-dimensional map. Such a map could be described by a particle representation as

$$\mathbf{x}(\hat{\mathbf{x}}, t) = f(\hat{\mathbf{x}}, t) = \sum_{j=1}^{M} \boldsymbol{\chi}(t)_j \varphi_j(\hat{\mathbf{x}}). \tag{3.1}$$

In the present method we introduce such a map, and we require that the basis functions $C^2(\hat{\Omega}) \ni \varphi_j(\hat{\mathbf{x}}) = \varphi(\hat{\mathbf{x}} - \boldsymbol{\xi}_j)$ be positive and have local support. Positivity is desirable, as it assures that monotonicity of the nodes $\{\boldsymbol{\xi}_j\}_{j=1}^M$ and node values $\{\boldsymbol{\chi}_j\}_{j=1}^M$ leads to monotonicity of the map. The parameters in the map that are changed in the process of adaptation are the node values $\{\boldsymbol{\chi}_j\}_{j=1}^M$. Due to the lack of simple invertibility we require that $\boldsymbol{\chi}_j(t) \in C^1([0,T])$ for all $j = 1, \ldots, M$, i.e., that the adaptation of the nodes be continuous and differentiable. Using a map as described in (3.1) makes it impossible to leap back and forth from physical to reference space. However, its differentiability enables us to cast the governing equations into reference space and solve the problem there, without the need of the inverse map $\mathbf{g} : \mathbf{x} \to \hat{\mathbf{x}}$, i.e., without the need of invertibility.

**3.1. Adaptive global mappings-based algorithm for a 1D convection-diffusion equation.** By virtue of the $C^1$ regularity of the map adaptation, we can express the governing equations of the problem in reference space. We shall illustrate this by considering the 1D problem (2.1) as an example.

Let $\mathbb{R} \supseteq \Omega \equiv \hat{\Omega} = [0,1]$, $f : \hat{x} \mapsto x$ be our map with $f \in C^1([0;T]) \times C^2(\hat{\Omega})$, metric $\Phi = \frac{\partial \hat{x}}{\partial x}$, $u : \Omega \times [0,T] \to \mathbb{R}$, and $q : \Omega \times [0,T] \to \mathbb{R}$ governed by

$$\frac{\partial q}{\partial t} + \frac{\partial (uq)}{\partial x} = \nu \frac{\partial^2 q}{\partial x^2}, \tag{3.2}$$

and assume periodic boundary conditions without loss of generality. Given an arbitrary map adaptation with the above-described regularity properties, we set

$$\mathcal{U} \equiv \frac{\partial f}{\partial t} = \sum_j^M \frac{d\chi_j}{dt}(t)\varphi_j(\hat{x}), \tag{3.3}$$

with

$$\hat{q}(\hat{x}(t),t) = q(f(\hat{x}(t),t),t) \tag{3.4}$$

and with

$$\frac{\partial q}{\partial t} = \frac{\partial \hat{q}}{\partial t} - \Phi \frac{\partial \hat{q}}{\partial \hat{x}} \mathcal{U}, \tag{3.5}$$

$$\frac{\partial q}{\partial x} = \Phi \frac{\partial \hat{q}}{\partial \hat{x}}, \tag{3.6}$$

$$\frac{\partial^2 q}{\partial x^2} = \Phi \frac{\partial}{\partial \hat{x}} \left( \Phi \frac{\partial \hat{q}}{\partial \hat{x}} \right), \tag{3.7}$$

and the problem (3.2) can be cast into reference space, yielding

$$\frac{\partial \hat{q}}{\partial t} - \Phi \mathcal{U} \frac{\partial \hat{q}}{\partial \hat{x}} + \Phi \frac{\partial (\hat{u}\hat{q})}{\partial \hat{x}} = \nu \Phi \frac{\partial}{\partial \hat{x}} \left( \Phi \frac{\partial \hat{q}}{\partial \hat{x}} \right). \tag{3.8}$$

In order to get (3.8) into conservative form we divide it by $\Phi$, and realizing that

$$\frac{\partial}{\partial t} \left( \frac{\hat{q}}{\Phi} \right) = \frac{1}{\Phi} \frac{\partial \hat{q}}{\partial t} + \hat{q} \frac{\partial \mathcal{U}}{\partial \hat{x}}, \tag{3.9}$$

we can write (3.8) as

$$\frac{\partial \hat{q}'}{\partial t} + \frac{\partial (\hat{q}' \tilde{u})}{\partial \hat{x}} = \nu \frac{\partial}{\partial \hat{x}} \left( \Phi \frac{\partial (\Phi \hat{q}')}{\partial \hat{x}} \right), \tag{3.10}$$

with $\hat{q}' \equiv \hat{q}/\Phi$ and mapped transport velocity

$$\tilde{u} = \Phi(\hat{u} - \mathcal{U}). \tag{3.11}$$

Now, with

$$\hat{Q} = \int_{\hat{V}} \hat{q}' d\hat{x} \quad \text{and} \quad \hat{v} \equiv |\hat{V}|,$$

we rewrite (3.10) in Lagrangian fashion as

$$\begin{aligned}\frac{d\hat{x}}{dt} &= \tilde{u},\\ \frac{d\hat{q}'}{dt} &= \nu\frac{\partial}{\partial\hat{x}}\left(\Phi\frac{\partial(\Phi\hat{q}')}{\partial\hat{x}}\right),\\ \frac{d\chi_i}{dt} &= \mathcal{U}_i = \mathcal{U}(\xi_i), \quad i=1,\dots,M,\end{aligned} \tag{3.12}$$

which solves (3.2). We now have two convective elements in this system: convection of the map (adaptation) with arbitrary velocity $\mathcal{U}(\hat{x},t)$ and convection of the particles with velocity $u$ in physical and $\Phi(\hat{u}-\mathcal{U})$ in reference space. With

$$\hat{Q}_j = \int_{\hat{V}_j} \hat{q}'\, d\hat{x}$$

and $\hat{v}_j \equiv |\hat{V}_j|$ we can discretize the above system using $N$ particles initially uniformly distributed in $\hat{\Omega}$ with positions $\hat{x}_j = j\Delta\hat{x}$, $\Delta\hat{x} = \frac{|\hat{\Omega}|}{N}$, and strengths $\hat{Q}_j$ for $j=1,\dots,N$. The particle solution to the transport problem (3.10) is obtained by numerically integrating the following ODEs:

$$\begin{aligned}\frac{d\hat{x}_j}{dt} &= \tilde{u}_j,\\ \frac{d\hat{Q}_j}{dt} &= \frac{\nu}{\hat{\epsilon}^2}\sum_k \frac{\Phi_k+\Phi_j}{2}[\hat{Q}_k\hat{v}_j\Phi_k - \hat{Q}_j\hat{v}_k\Phi_j]\eta^{\hat{\epsilon}}(\hat{x}_j-\hat{x}_k),\\ \frac{d\hat{v}_j}{dt} &= \frac{1}{\hat{\epsilon}}\sum_k[\tilde{u}_j+\tilde{u}_k]\hat{\nabla}\zeta^{\hat{\epsilon}}\hat{v}_k\hat{v}_j,\\ \frac{d\chi_i}{dt} &= \mathcal{U}(\xi_i), \quad i=1,\dots,M,\end{aligned} \tag{3.13}$$

where $\hat{v}_j$ is the volume of the particles which is usually initialized as $\hat{v}_j(0)=\Delta\hat{x}$ and we have set $[\cdot]_j \equiv [\cdot](\hat{x}_j)$ to keep things short.

**3.2. Multidimensional case.** The equations for a multidimensional convection-diffusion equation can be found using similar derivations [19].

Let $\mathbb{R}^d \supseteq \Omega \equiv \hat{\Omega} = [0,1]^d$, $\mathbf{f}:\hat{\mathbf{x}}\mapsto\mathbf{x}$ be the map now with $\mathbf{f}\in C^1([0,T])\times C^2(\hat{\Omega})$, metric tensor $\{\underline{\boldsymbol{\Phi}}\}_{ij} = \frac{\partial\hat{x}_i}{\partial x_j}$, $|\underline{\boldsymbol{\Phi}}|$ its determinant, $\mathbf{u}:\Omega\times[0,T]\to\mathbb{R}^d$, and $q:\Omega\times[0,T]\to\mathbb{R}$ governed by

$$\frac{\partial q}{\partial t} + \nabla\cdot(\mathbf{u}q) = \nu\Delta q, \tag{3.14}$$

again with periodic boundary conditions, and for simplicity we assume that $\nabla\cdot\mathbf{u}=0$. With the map adaptation velocity given as

$$\boldsymbol{\mathcal{U}} = \sum_j^M \frac{d\boldsymbol{\chi}_j}{dt}(t)\phi_j(\hat{\mathbf{x}}), \tag{3.15}$$

and with

$$\hat{q}' = (|\underline{\boldsymbol{\Phi}}|)^{-1}\hat{q}, \tag{3.16}$$

the convection-diffusion equation (3.14) becomes

$$\frac{\partial \hat{q}'}{\partial t} + \hat{\nabla} \cdot (\tilde{\mathbf{u}} \hat{q}') = \nu \frac{\partial}{\partial \hat{x}_i} \left( b_{ij} \frac{\partial}{\partial \hat{x}_j} (|\underline{\mathbf{\Phi}}| \hat{q}') \right), \tag{3.17}$$

with $b_{ij}$ from (2.10). The composite convection velocity in reference space $\tilde{\mathbf{u}}$ is given by

$$\tilde{\mathbf{u}} = \underline{\mathbf{\Phi}} (\hat{\mathbf{u}} - \boldsymbol{\mathcal{U}}). \tag{3.18}$$

This leads to the following set of ODEs for the particle locations, strengths, and map-nodes:

$$\begin{aligned} \frac{d\hat{\mathbf{x}}_j}{dt} &= \tilde{\mathbf{u}}_j, \\ \frac{d\hat{Q}_j}{dt} &= \frac{\nu}{\hat{\epsilon}^2} \sum_k \psi^{\hat{\epsilon}}_{pq}(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_k) \left( \frac{m_{pq}(\hat{\mathbf{x}}_j) + m_{pq}(\hat{\mathbf{x}}_k)}{2} \right) [\hat{v}_j \hat{Q}_k |\underline{\mathbf{\Phi}}|_k - \hat{v}_k \hat{Q}_j |\underline{\mathbf{\Phi}}|_j], \\ \frac{d\boldsymbol{\chi}_i}{dt} &= \boldsymbol{\mathcal{U}}(\boldsymbol{\xi}_i), \quad i = 1, \dots, M. \end{aligned} \tag{3.19}$$

**3.3. Choice of map adaptation.** We limit our presentation to convection-dominated problems for which particle methods are primarily used. There the map adaptation can be chosen based on physical reasoning, which we will illustrate for the 1D case, or by a numerical procedure such as the moving mesh partial differential equations (MMPDEs) for the adaptation velocity.

For convection-dominated problems dissipative forces are typically small, and a reasonable suggestion for the map adaptation $\mathcal{U}$ is

$$\mathcal{U} \sim \hat{u}. \tag{3.20}$$

This choice has the effect that particles give in to the deformation forces acting upon them; from $\mathcal{U} \sim \hat{u}$ it follows that

$$\frac{d\epsilon}{dt} = \frac{d}{dt} \left( \hat{\epsilon} \frac{1}{\Phi} \right) = \hat{\epsilon} \frac{\partial \mathcal{U}}{\partial \hat{x}} \sim \epsilon \frac{\partial u}{\partial x},$$

which means that the core size in the physical space is being deformed in the same way the volume is deformed by the flow. This relation between volume deformation and particle deformation does not exist in nonadaptive particle methods. Of course, in dimensions higher than one, if the flow is incompressible, the magnitude of the volume does not adapt, but merely its shape gives in to the deformation.

As r-adaptive methods do not perform an actual scale decomposition like, for instance, multiresolution-analysis-based methods or the AMR-based method described hereafter do, they cannot remove the "multiscale" character from a problem. Nonetheless, r-adaptivity can distribute the multiscale character between the solution of the problem in reference space $\hat{q}'$ and the map $f$. The question of an optimal map—in terms of minimizing the approximation error in a given norm—is deeply hidden in the approximation error of the function approximation and the approximated operators. The choice (3.20) for the adaptation is made based on physical reasoning and is not guaranteed to minimize discretization errors. Furthermore, as the conservation of volume implies that if a particle volume is compressed, other particles must be expanded,

there is a point where increasing the adaptation will lead to a bigger error, especially if a number of particles is used which is able to fully resolve the finest length scales of the problem without adaptivity. Although in higher dimensions the particles are typically not compressible, the statement is still valid, as too high a degree of anisotropy will also lead to increased error. This "sharing" of the multiscale character has the effect that if we set $\mathcal{U} = \hat{u}$, convection will not generate any small scales in $\hat{q}'(\hat{x}, t)$, but the map will become increasingly multiscale, which in turn will lead to increased error. We therefore set

$$\mathcal{U} = \gamma \hat{u}, \tag{3.21}$$

where $\gamma \geq 0$ is a robust parameter of the method which typically would be chosen small if $\frac{\Delta \hat{x}}{\delta_{\text{phys}}}$ is small, $\delta_{\text{phys}}$ being the smallest characteristic length scale in the flow.

In the multidimensional case the choice (3.21) will lead to an increasingly anisotropic map $\mathbf{f}$, and accurate computation of the metric $\underline{\mathbf{\Phi}}$ becomes difficult. For this case we suggest a different strategy: we require that the map adaptation velocity satisfies a MMPDE, a technique commonly used in grid-based r-adaptive methods. The MMPDEs can be derived by requiring equidistribution of a monitor function which determines where in the computational domain an increased amount of map nodes is required. Specifically, we adopt the methodology introduced in [3] and set

$$\boldsymbol{\mathcal{U}} \sim \hat{\nabla} \cdot M(\hat{\mathbf{x}}, t) \hat{\nabla} (\mathbf{f}(\hat{\mathbf{x}}, t))^{\mathsf{T}}. \tag{3.22}$$

The monitor function $M(\hat{\mathbf{x}}, t)$ is a positive measure which takes great values where numerical resolution should be increased and is usually expressed as

$$M(\hat{\mathbf{x}}, t) = \sqrt{1 + \alpha |\mathcal{D}^m \hat{q}'|^2}, \tag{3.23}$$

where $\mathcal{D}^m$ is a differential operator in $\hat{\Omega}$ of order $|m| \geq 0$, e.g., $M = \sqrt{1 + \alpha |\hat{\nabla} \hat{q}'|^2}$, to resolve steep gradients of $\hat{q}'$ and the adaptation parameter $\alpha$ takes the role of the parameter $\gamma$ above. We refer the reader to [3] for a more detailed derivation.

**3.4. Map regularization.** If there are no dissipative forces present in the adaptation of the map, it is necessary to actively ascertain that the regularity of the map is maintained. Our numerical experiments suggest that the method can handle great values of $\underline{\mathbf{\Phi}}$—i.e., great particle compression ratios—without becoming numerically inaccurate or unstable. But especially when adaptation acts very locally, that is, in regions where $|\hat{\nabla}(|\underline{\mathbf{\Phi}}|)|$ is big, it is necessary to keep the map smooth. As in most methods that are based on r-adaptivity we alleviate this problem by continuously smoothing the map in an adaptive manner: in one dimension we replace the map $\{\boldsymbol{\chi}_j\}_{j=1}^M$ at every time step by its regularization $\{\tilde{\boldsymbol{\chi}}_j\}_{j=1}^M$, computed as

$$\tilde{\chi}_j = \frac{\chi_j + |\kappa \frac{\partial \Phi}{\partial \hat{x}}|^2 (\chi_{j-1} + \chi_{j+1})}{1 + 2|\kappa \frac{\partial \Phi}{\partial \hat{x}}|^2}, \tag{3.24}$$

with $0 < \kappa \ll 1$ being a smoothing parameter that can be chosen independent of $M$ or $N$. In two dimensions this regularization takes an according shape; assuming that the map is defined on a grid and node $j$ corresponds to grid point $m, n$,

$$\tilde{\boldsymbol{\chi}}_{m,n} = \frac{\boldsymbol{\chi}_{m,n} + |\kappa \hat{\nabla}(|\underline{\mathbf{\Phi}}|)|^2 (\boldsymbol{\chi}_{m,n-1} + \boldsymbol{\chi}_{m,n+1} + \boldsymbol{\chi}_{m-1,n} + \boldsymbol{\chi}_{m+1,n})}{1 + 4|\kappa \hat{\nabla}(|\underline{\mathbf{\Phi}}|)|^2}. \tag{3.25}$$

This smoothing can also be seen as a diffusion step with a diffusion constant

$$\nu_r = \frac{1}{\Delta t}(\Delta\xi|\kappa\hat{\nabla}(|\underline{\mathbf{\Phi}}|)|)^2,$$

$\Delta\xi$ being the map node spacing. Clearly, after each such regularization step, the adaptation velocity $\boldsymbol{\mathcal{U}}$ must be corrected by the amount the $\boldsymbol{\chi}_j$ have moved due to regularization.

**4. Particle method with AMR.** The method we describe here is also in the spirit of the variable-blob method [6]; but while the method presented above is based on an adaptive global map, we employ a combination of simple local mappings.

In [6] different mappings were used in different parts of the computational domain, leading to different grid sizes. Typically, for a flow around obstacles, one can think of local mappings adapted to a fine resolution of each boundary layer, while another mapping could be used in intermediate zones with a stretched resolution away from the obstacles. These methods can be viewed as nonconforming domain decomposition methods with domain overlapping.

We propose here a method along the same lines but with piecewise constant grid sizes adaptively adjusting to the solution. At every time step blocks with grid sizes, say, of the form $h2^{-l}$ are defined and discretized by particles with corresponding blob sizes. As in the method proposed in [6], the overlapping of the blocks is essential to allow particles around the block interfaces to exchange information and maintain a consistent approximation at the desired resolution everywhere. The exchange of information is done by interpolation at the remeshing stage that in our particle algorithms is done at every time step. In what follows we detail this procedure for a 1D advection problem; then we describe how to handle diffusion and variable time steps.

**4.1. AMR-based algorithm for a 1D advection equation.** To describe more precisely the algorithm, we focus on (2.1) and consider the case of two given blocks $\Omega_c$ and $\Omega_f$:

$$\Omega_c = ]-\infty, a[, \qquad \Omega_f = ]b, +\infty[,$$

with $a > b$. $\Omega_c$ and $\Omega_f$ are, respectively, coarse and fine resolution zones, with particle sizes $H$ and $h$.

We will denote by $x^n_{c,i}$ (resp., $x^n_{f,i}$) the locations of particles in the coarse (resp., fine) block at time $t_n = n\Delta t$ and by $Q^n_{c,i}$ (resp., $Q^n_{f,i}$) their strengths. A complete time step of the algorithm alternates particle motion and remeshing. Particles are pushed in both zones in the same way as in a single-resolution method. We assume that after particles have moved they provide a consistent approximation of the solution $q$, at the corresponding resolution, in both domains. We denote by $\tilde{x}^{n+1}_{c,i}$ and $\tilde{Q}^{n+1}_{c,i}$ the particle locations and weights after advection in the coarse-grid domain, with similar notation in the fine-resolution domain. The goal of the remeshing step that follows is to make sure this assumption will still be valid at the next time step. For this purpose we need to extend $\Omega_c$ further inside $\Omega_f$, and vice versa. We thus define

$$\overline{\Omega}_c = ]-\infty, a + l_1[, \qquad \overline{\Omega}_f = ]b - l_2, +\infty[.$$

We also set

$$\widetilde{\Omega}_c = ]-\infty, a - l_3[, \qquad \widetilde{\Omega}_f = ]b + l_4, +\infty[,$$

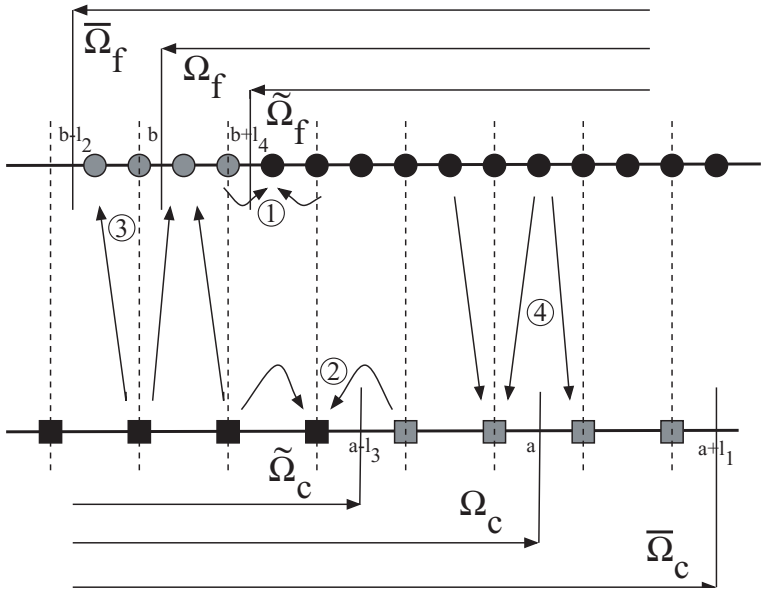


FIG. 2. *Sketch of coarse-fine domain decomposition. Arrows indicate how particles at the end of advection contribute to remeshed particle values in the different domains. Particles in grey (resp., in black) obtain their strength after remeshing from the domain with a different resolution (resp., the same resolution). Numbers refer to the different stages in the remeshing algorithm.*

with $l_i > 0$ such that $a - l_3 > b + l_4$, so that $\widetilde{\Omega}_c$ and $\widetilde{\Omega}_f$ overlap (see the sketch on Figure 2). The remeshing step will remesh existing particles $\tilde{x}_{c,i}^{n+1}$, $\tilde{x}_{f,i}^{n+1}$ at regular locations $x_{c,i}^{n+1}$, $x_{f,i}^{n+1}$ (resp., in $\widetilde{\Omega}_f$ and $\widetilde{\Omega}_c$) and create new particles in $\overline{\Omega}_c - \widetilde{\Omega}_c$ and $\overline{\Omega}_f - \widetilde{\Omega}_f$. More precisely, the remeshing algorithm proceeds as follows:

1. Particle weights of fine-size particles in $\Omega_f$ are interpolated to give values for particles at regular locations on a fine grid in $\widetilde{\Omega}_f$.

2. Particle weights of coarse-size particles in $\Omega_c$ are interpolated to give values for particles at regular locations on a coarse grid in $\widetilde{\Omega}_c$.

3. Coarse-size particles in $\widetilde{\Omega}_c$ are used to compute values for particles at regular locations on a fine grid in $\overline{\Omega}_f - \widetilde{\Omega}_f$

4. Fine-size particles in $\widetilde{\Omega}_f$ are used to compute values for particles at regular locations on a coarse grid in $\overline{\Omega}_c - \widetilde{\Omega}_c$

The remeshing in steps 1, 2, and 3 above are done by interpolation with the $M_4'$ kernel. Steps 3 and 4 can either be done simultaneously with steps 1 and 2 or follow these stages. In that case, step 4 is just a sampling of the values obtained in step 2.

Some remarks are in order to clarify these steps and to check their consistency. First, it is important to observe that the fine-size particles in $\widetilde{\Omega}_f$ do not suffice to give consistent remeshed values throughout $\widetilde{\Omega}_f$, unless a one-sided interpolation formula was used near the domain interfaces. We did not consider that option, as it would add some algorithmic complexity. On the other hand, using information from the coarse-grid domain, as prescribed in step 3, is consistent, provided that the stencil needed for the interpolation remains in $\Omega_c$. Similarly, for the remeshing from $\Omega_f$ to $\widetilde{\Omega}_f$ to be consistent, we need the stencil centered at the boundary of $\widetilde{\Omega}_f$ to not extend outside $\Omega_f$. These observations also give us overlapping rules that the domains $\Omega$, $\widetilde{\Omega}$,

and $\overline{\Omega}$ must obey. Let $k$ be the number of points in each direction involved in the interpolation function used to remesh particles ($k = 2$) for the $M_4'$ used here.

- Consistency of step 1 above requires $l_4 \geq (k-1)h$.
- Consistency of step 2 above requires $l_3 \geq (k-1)H$.
- Consistency of step 3 requires that $b - l_4 + kH \in \Omega_c$, that is, $a - b - l_4 > kH$.
- Consistency of step 4 requires that $a - l_3 - kh \in \Omega_f$, that is, $a - b - l_3 > kh$.

Denoting by $\delta = b - a$ the width of overlapping between the two domains, these conditions can be rewritten as

$$hk \leq l_4 < \delta - kH, \qquad Hk \leq l_3 < \delta - kh. \tag{4.1}$$

We deduce from these conditions that $\delta$ must satisfy

$$\delta > k(H + h). \tag{4.2}$$

Under conditions (4.1) and (4.2), given the fact that particles $\tilde{x}_{c,i}^{n+1}$ and $\tilde{x}_{f,i}^{n+1}$ with weights $\tilde{Q}_{c,i}^{n+1}$ and $\tilde{Q}_{f,i}^{n+1}$ were a consistent discretization of $\Omega_c$ and $\Omega_f$, remeshed particles $x_{c,i}^{n+1}$ and $x_{f,i}^{n+1}$ with weights $Q_{c,i}^{n+1}$ and $Q_{f,i}^{n+1}$ provide a consistent discretization of $\overline{\Omega}_c$ and $\overline{\Omega}_f$ at the corresponding resolution. It remains then to ensure that the next advection step will keep consistent particle values in $\Omega_c$ and $\Omega_f$. If $\|u\|_\infty$ denotes the maximum advection velocity, a sufficient condition is clearly

$$l_1 \geq \|u\|_\infty \Delta t, \qquad l_2 \geq \|u\|_\infty \Delta t. \tag{4.3}$$

**4.2. Diffusion, domain adaptation, and variable time step.** Handling a diffusion term does not pose additional difficulties. In our particle algorithms, diffusion is solved at the end of the remeshing step, by a PSE scheme with discrete normalization as discussed in section 2.2, using particles on a finite-size stencil. Second order, three-point stencils are routinely used. In that case, diffusion returns consistent approximations in the domains $\overline{\Omega}_c$ and $\overline{\Omega}_f$, except for the last particles. As a result, conditions (4.3) on $l_1$ and $l_2$ for the particles $Q_{c,i}^{n+1}$ and $Q_{f,i}^{n+1}$ to yield consistent approximations after convection-diffusion have to be modified into

$$l_1 - H \geq \|u\|_\infty \Delta t, \qquad l_2 - h \geq \|u\|_\infty \Delta t. \tag{4.4}$$

Condition (4.4) assumes that the domains do not change in time. In a fully adaptive method, coarse and fine blocks are updated at every time step, and this must be accounted for in the choice of $l_2$ and $l_4$: if $\Omega_{c,f}^n$ and $\Omega_{c,f}^{n+1}$ denote fine and coarse blocks at times $t_n$ and $t_{n+1}$, and if $\lambda$ is a positive number such that

$$d(\Omega_c^n, \Omega_c^{n+1}) \leq \lambda, \qquad d(\Omega_f^n, \Omega_f^{n+1}) \leq \lambda,$$

then condition (4.4) becomes

$$l_2 - h - \geq \lambda + \|u\|_\infty \Delta t, \qquad l_1 - H \geq \lambda + \|u\|_\infty \Delta t. \tag{4.5}$$

In our implementations of this method we have chosen to adapt the resolution such that it obeys a criterion based on the gradients. In all cases it is possible to estimate at every time step $\lambda$ in terms of $\Delta t$ and the coefficient $u$ and its derivatives.

As already mentioned in section 2, in particle methods for advection equations the choice of the time step is dictated by the need to prevent particle paths from crossing. This gives the condition

$$\Delta t \leq \frac{C}{\max |\partial u / \partial x|}, \tag{4.6}$$

where $C$ depends on the time-stepping scheme.

For nonlinear problems, the velocity is part of the computation, and it is likely that fine-size particles will be used in zones where $|\partial u/\partial x|$ is large. It is therefore desirable to be able to adapt the time step accordingly. In our algorithm, this option turns out to be rather simple to implement. Assume that $\Delta t_c$ and $\Delta t_f$ are the time steps used, respectively, in $\overline{\Omega}_c$ and $\overline{\Omega}_f$, for instance based on (4.6), with

$$\Delta t_c > \Delta t_f, \quad \Delta t_c/\Delta t_f = K \in \mathbb{N}. \tag{4.7}$$

Our multilevel method can be implemented as follows, where the subscript refers to iterations at the coarse-grid level and, for simplicity, we assumed that the coarse and fine grid domains do not change in time: coarse-size particles $x^n_{c,i}$, originating in $\overline{\Omega}^n_c$, are advected with their local velocities for a time step $\Delta t_c$ to give locations $\tilde{x}^{n+1}_{c,i}$. By linear interpolation, particles locations are determined at intermediate time steps corresponding to multiples of $\Delta t_f$. We denote by $\tilde{x}^{n+l/K}_{c,i}$ these locations for $1 \le l \le K$.

Simultaneously, we advance fine-size particles originating in $\overline{\Omega}_f$ with time step $\Delta t_f$. Remeshing of the particles proceeds as follows. At the end of the $K-1$ first fine time steps, we use coarse-size particles $\tilde{x}^{n+l/K}_{c,i}$ and fine-size particles $\tilde{x}^{n+l/K}_{f,i}$ to remesh fresh particles in $\overline{\Omega}_f$ (steps 1 and 3 in the remeshing algorithm) that are subsequently advected for the next substep $\Delta t_f$. After $K$ substeps, we finally use both fine- and coarse-size particles $\tilde{x}^{n+1}_{c,i}$ and $\tilde{x}^{n+1}_{f,i}$ in steps 2 and 4 of the remeshing algorithm to get coarse-size particles at regular locations in $\overline{\Omega}_c$.

In a particle method, a substantial part of the computational time is devoted to the interpolation formulas in the remeshing steps (in three dimensions, the computational complexity of this step is of order $64N$ for $N$ particles). By adapting the remeshing frequency to the local strain in the flow, the present method allows us to keep this effort to its minimal requirement.

## 5. Numerical illustrations.

**5.1. 1D examples.** To illustrate the two adaptive methods described above, we have focused on the viscous Burgers equation as a test case. In what follows, to give an idea of the overall efficiency of the particle schemes that are underlying our methods for hyperbolic conservation laws, we first show comparisons of these schemes with classical ENO schemes in the case of uniform grids. We then turn to our adaptive methods.

We consider the following nonlinear conservation law: $u : [0,1] \equiv \Omega \times [0,T] \to \mathbb{R}$,

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} &= \nu \frac{\partial^2 u}{\partial x^2}, \\ u(x,0) &= u_o(x), \\ u(0,\cdot) &= u(1,\cdot), \end{aligned} \tag{5.1}$$

where $\nu = 0$ for the inviscid case and $\nu = 10^{-3}$ for the viscous one. We have used two types of initial conditions: a simple sine leading to a steady shock,

$$u_o(x) = \sin(2\pi x), \tag{5.2}$$

and an asymmetric function leading to a moving shock,

$$u_o(x) = \sin(2\pi x) + \frac{1}{2}(1 - \cos(2\pi x)). \tag{5.3}$$

**5.1.1. Inviscid case.** Following section 2, the particle approximation of above system consists of writing

$$u(x,t) \simeq \sum_j v_j u_j \zeta^\epsilon(x - x_j). \tag{5.4}$$

Setting $\Gamma_j = v_j u_j$, (2.6) yields the following set of ODEs:

$$\frac{dx_j}{dt} = u_j, \tag{5.5}$$

$$\frac{d\Gamma_j}{dt} = \frac{\nu}{\epsilon^2} \sum_k [v_j \Gamma_k - v_k \Gamma_j] \eta^\epsilon(x_j - x_k), \tag{5.6}$$

$$\frac{dv_j}{dt} = \frac{1}{\epsilon} \sum_k [u_j + u_k] \hat{\nabla} \zeta^\epsilon(x_j - x_k) v_k v_j. \tag{5.7}$$

In our implementation of the method, each time step is made of a Runge–Kutta iteration for the motion of particles followed by a remeshing and by the diffusion. In that case, diffusion is done on particles lying on a uniform grid. Thus (5.6) can be reduced, for a piecewise linear kernel $\eta$, to a classical three-point finite difference scheme. Likewise, for the motion of particles (5.5) and (5.7) with a second order Runge–Kutta scheme, half a step is done to advance volumes; then velocities are recovered by $\Gamma_j / v_j$, and particles are pushed with these velocities for a time step. Since at the beginning of the time step particles are on a uniform grid, the formula (5.7) also reduces to a standard finite difference formula. Finally, our numerical discretization was completed by an artificial viscosity model [4], which is based on an analysis of the discretization error in particle methods. For the Burgers equation, it consists of a nonlinear diffusion of von Neumann type acting only in compression zones:

$$\frac{d\Gamma_j}{dt} = C \sum_k \left[ (u_j - u_k) \cdot \frac{(x_j - x_k)}{|x_j - x_k|^2} \right]_- [v_j \Gamma_k - v_k \Gamma_j] \eta^\epsilon(x_j - x_k). \tag{5.8}$$

Again, since diffusion acts on particles lying on a uniform grid, the above formula can be implemented as a finite difference three-point formula. To illustrate the accuracy of the method, we show comparisons with third order weighted ENO schemes [18].

Figure 3 is a comparison for low resolution ($N = 25$ grid points or particles) of the particle method with third order WENO schemes based on either Lax–Friedrichs or Engquist–Osher flux evaluations at time 0.2 for these initial conditions. Although particle methods could use larger time steps, for simplicity the same time step was used for all simulations, viscosity was set to 0, and the coefficient $C$ in the artificial viscosity model (5.8) was set to 0.5. Figure 4 shows discrete relative $L^2$ norms of the error in a refinement study based on the second initial condition for the WENO and particle schemes. A tentative conclusion of these tests is that particle methods as we implement them can be considered in terms of accuracy for shock capturing as intermediate between third order WENO schemes based on Lax–Friedrichs and Engquist–Osher numerical fluxes.

**5.1.2. Global adaptive mapping-based method.** Setting $\hat{\Gamma}_j = \frac{\hat{u}_j}{\Phi_j} \hat{v}_j$ we readily obtain the discretization of the viscous version ($\nu = 10^{-3}$) of (5.1) for

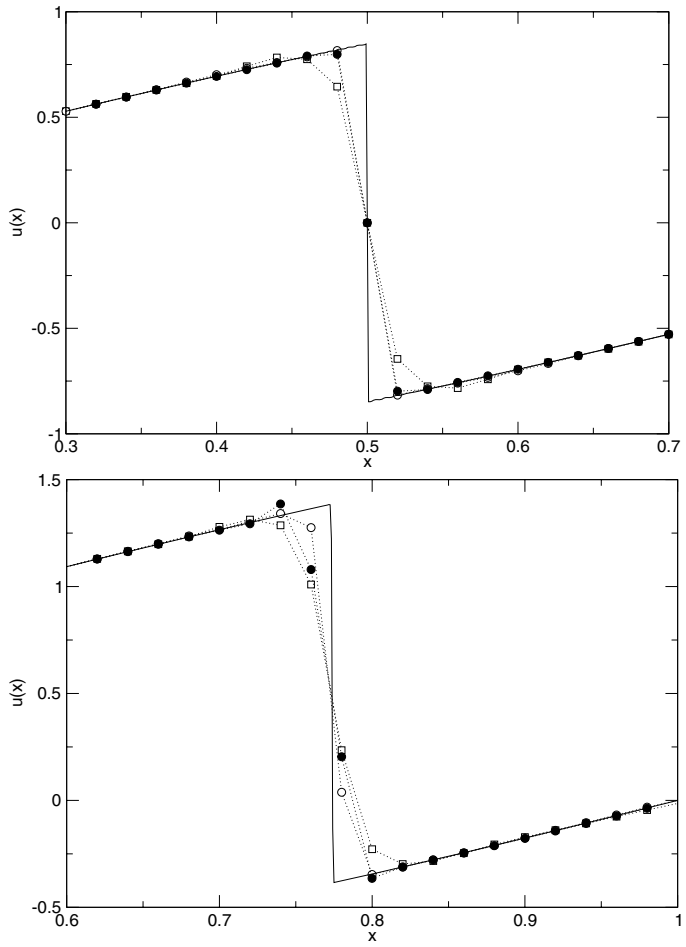


FIG. 3. *Comparison of third order WENO schemes using the Lax–Friedrichs flux (▫··▫, white squares), the Engquist–Osher flux (∘·∘, white circles) and the nonadaptive particle method (•··•, black circles) with initial condition* (5.2) *(upper picture) and* (5.3) *(lower picture).*

$j = 1, \ldots, N$:

$$\frac{d\hat{x}_j}{dt} = \tilde{u}_j, \tag{5.9}$$

$$\frac{d\hat{\Gamma}_j}{dt} = \frac{\nu}{\hat{\epsilon}} \sum_k \frac{\Phi_k + \Phi_j}{2} [\hat{\Gamma}_k \hat{v}_j \Phi_k - \hat{\Gamma}_j \hat{v}_k \Phi_j] \eta^{\hat{\epsilon}}(\hat{x}_j - \hat{x}_k), \tag{5.10}$$

$$\frac{d\hat{v}_j}{dt} = \frac{1}{\hat{\epsilon}} \sum_k [\tilde{u}_j + \tilde{u}_k] \nabla \zeta^{\hat{\epsilon}} \hat{v}_k \hat{v}_j, \tag{5.11}$$

where again

$$\tilde{u}_j = \Phi_j \left( \frac{\Phi_j}{\hat{v}_j} \hat{\Gamma}_j - \mathcal{U}(\hat{x}_j) \right),$$

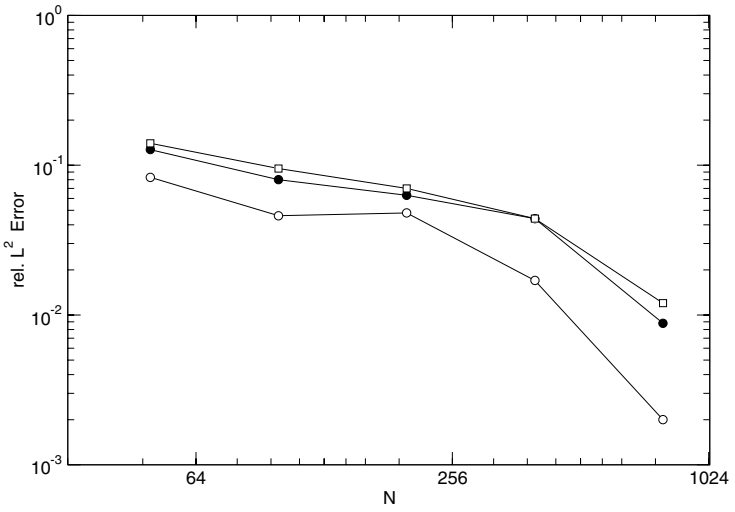


FIG. 4. *Refinement study for $N = 50$ to $N = 800$ points for WENO schemes using the Lax–Friedrichs flux (▫-▫, white squares), the Engquist–Osher flux (⊶, white circles), and the particle scheme (•—•, black circles) at time $t = 0.2$ with initial condition* (5.3).

and for $i = 1, \dots, M$ we have

$$\frac{d\chi_i}{dt} = \mathcal{U}(\xi_i) \tag{5.12}$$

$$\chi_i \leftarrow \frac{\chi_i + |\kappa \frac{\partial \Phi(\xi_i)}{\partial \hat{x}}|^2 (\chi_{i-1} + \chi_{i+1})}{1 + 2|\kappa \frac{\partial \Phi(\xi_i)}{\partial \hat{x}}|^2} \tag{5.13}$$

for the map. The same kernels and time-stepping technique as described in the inviscid case have been used for the particles and the map, with the only difference being that the map nodes have been pushed using two Euler steps of size $\frac{1}{2}\Delta t$ per time step. Additionally, we have not used an artificial viscosity model for this method—as it requires some modifications to (5.8) which have yet to be developed. The particles have been initialized on a uniform grid, and the map was initialized as the identity map, i.e.,

$$\chi_i = \xi_i, \quad i = 1, \dots, M.$$

We have used two different parameter setups in our experiments. In one the number of map discretization points $M$ was held constant, independent of the number of particles $N$ (`RADA2`), and in the other it was set equal to $N$ (`RADA1`). The scaling of the map adaptation velocity was set to be $\gamma = 0.7$ and the smoothing parameter $\kappa = 10^{-3}$, both independent of $N$ or $M$. These choices gave good results, but the fact that the same parameters were used for a broad range of $N$ emphasizes that they are indeed of robust nature, and we will not present any parameter studies in the following. We now present convergence studies of the method in the case of the moving "shock" (5.3) and compare against a nonadaptive scheme laid out as described in section 5.1.1 with artificial viscosity coefficient $C = 0.1$.

Although the system is viscous, a nonadaptive numerical solution with $N \lesssim 2000$ is still underresolved. Figure 5 shows the solutions at $t = 0.2$ with $N = 50$ for both

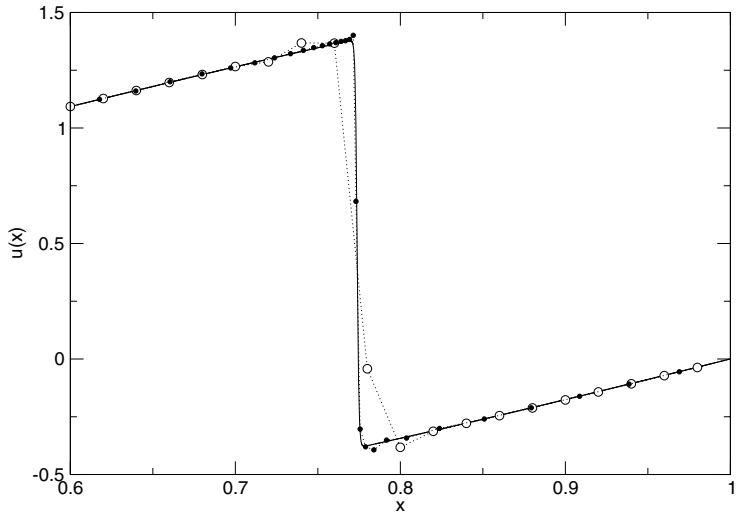


FIG. 5. *The reference solution (solid line) with* $N = 100,000$*; the adaptive solution using* `RADA1` *(•·•, black circles) and the nonadaptive solution (∘·∘, white circles), both with* $N = 50$ *at* $t = 0.2$.

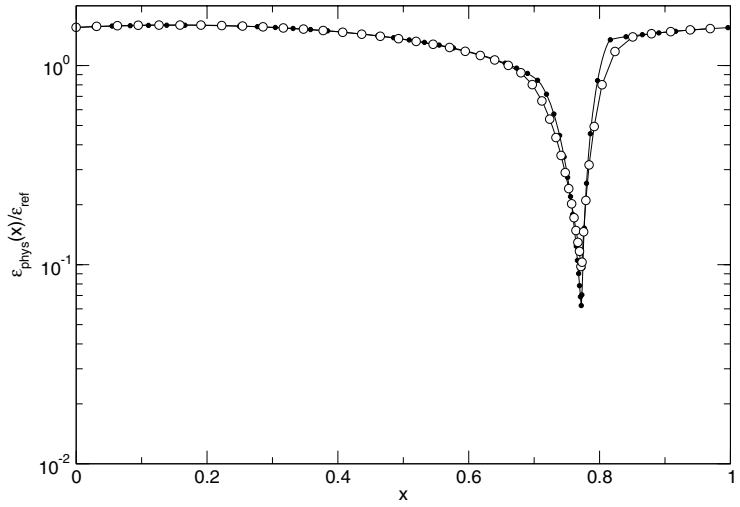


FIG. 6. *Distribution of core size ratio* $\epsilon(x)/\hat{\epsilon}$ *across the computational domain. Results are shown for* $N = 50$ *(⊶, white circles) and* $N = 400$ *(•—•, black circles), where only* $1/6$ *of the particles are shown for clarity.*

the adaptive (`RADA1`) and nonadaptive method. Figure 6 illustrates the adaptation of the particle core sizes across the computational domain at time $t = 0.2$. The ratio of biggest to smallest cores ranges up to one order of magnitude. This distribution has been taken from `RADA1` runs, and therefore the maximum amount of the ratio depends on the number of particles used, because as $M = N$ the resolution of the adaptation increases with $N$.

Figures 7 and 8 show the convergence behavior of `RADA1`, `RADA2`, and the nonadaptive scheme measured in discrete, relative $L^2(\Omega)$, and $H^1(\Omega)$ seminorms. The latter gives some idea of the resolvedness of the solution. The wiggles in the nonadaptive

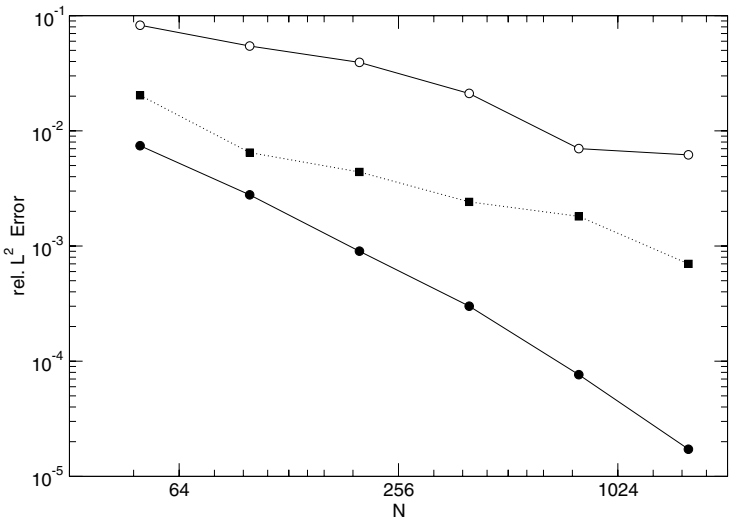


FIG. 7. *$L^2$ error versus number of particles $N$ for* `RADA1` *(•—•, black circles, solid line),* `RADA2` *(■··■, black squares, dotted line), and the nonadaptive method (○—○, white circles, solid line).*

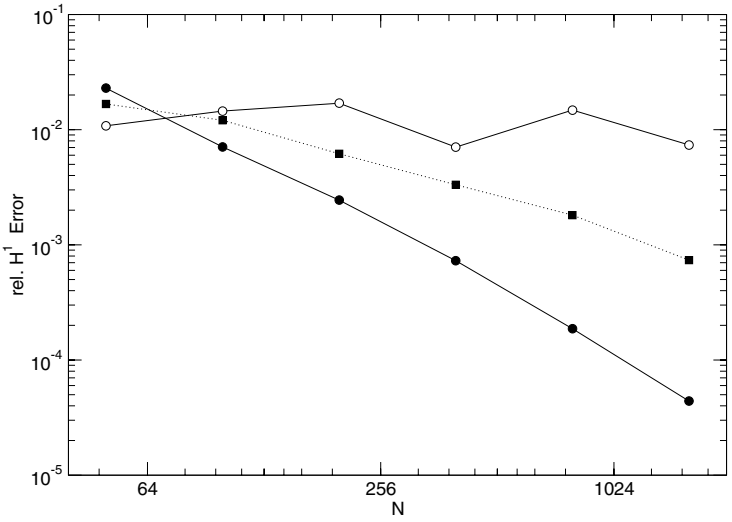


FIG. 8. *$H^1$-seminorm error versus number of particles $N$ for* `RADA1` *(•—•, black circles, solid line),* `RADA2` *(■··■, black squares, dotted line), and the nonadaptive method (○—○, white circles, solid line).*

case can be clearly connected to the nonresolution of the shock. Also the `RADA2` scheme shows some degree of irregularity of convergence. The reason for this behavior is the same as for the nonadaptive scheme: as the number of map nodes is kept constant and fairly small, the shock cannot be tracked as accurately as with `RADA1`. The greater convergence rate of `RADA1` is also expected: with growing number of particles $N$, not only the resolution capability of the particles is increased, but also the map is able to refine to smaller structures as the number of map nodes used increases accordingly.

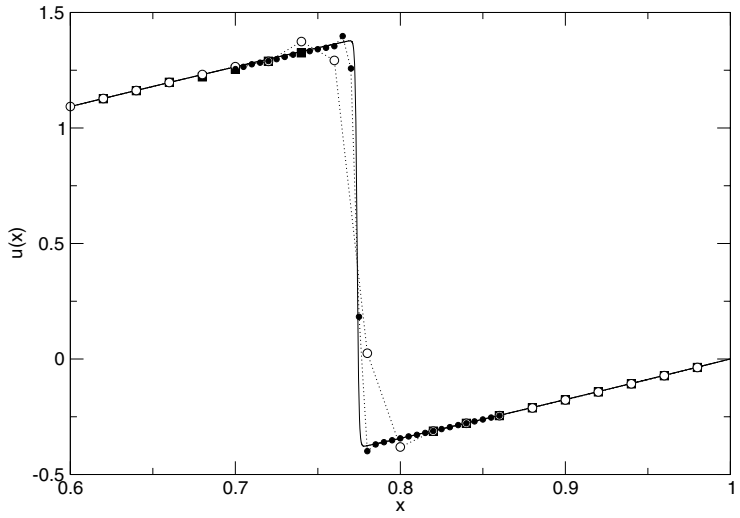


FIG. 9. *The reference solution (solid line), the nonadaptive particle solution (∘···∘, white circles, dotted line), and the adaptive particle solution (coarse domain part: ■···■, black squares, and fine domain part: •···•, black circles, dotted line) with degree of refinement* $H/h = 4$.

**5.1.3. AMR-based method.** The AMR particle method described in section 4 has been implemented for the Burgers equation with initial condition (5.3) and viscosity $\nu = 10^{-3}$. For both adaptive and nonadaptive methods, the artificial viscosity coefficient was set to 0.1. For that example the adaptation technique was designed to follow the shock: at every time step the location of the maximum velocity gradient is computed. Since in this computation the shock is established rather quickly, we designed the fine-resolution domain to extend to a constant number of coarse mesh sizes on each side of the shock. In the results that we show, the level of grid refinement, that is, the ratio $H/h$ in the notation of section 4, was varied from 2 to 20. The overlapping parameters were chosen according to (4.1) and (4.5). Figure 9 shows a comparison of the nonadaptive method and the adaptive method with $N = 50$ particles and the adaptive method with the same coarse resolution $H = 0.02$ and $H/h = 4$. The gain offered by the adaptive method in capturing the shock is clear.

In Figure 10 we have plotted the relative discrete error for the nonadaptive method and the adaptative method as a function of the number of particles, for the nonadaptive method, and of the number of particles that would correspond, in a nonadaptive method, to the finest interparticle spacing. For the adaptive method, the coarse interparticle spacing was kept constant with $H = 0.02$. Every point on the curve thus corresponds to a different ratio $H/h$, ranging from 2 to 20. The fine-resolution zone had a width of 3 coarse-grid sizes for $N = 50$ to $N = 500$; then this width has been increased to 10 coarse-grid sizes. The reason for that is that if the width of the fine zone is kept constant, beyond a certain level of refinement, the error in the coarse-grid zone becomes predominant. To avoid a saturation of the error level, it is thus necessary either to allow more than 1 level of refinement, a possibility that we have not implemented, or to increase the size of the fine-resolution domain.

One can observe that the two curves are very close. This proves that the adaptive method retains the overall accuracy of the fine-resolution domain. In other words, the interpolation involved at every time step around the interfaces to exchange

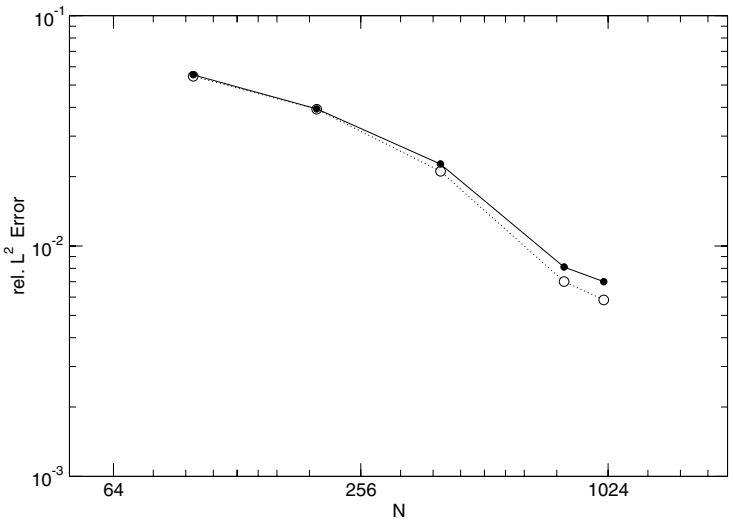


FIG. 10. *$L^2$ error reference study and comparison of the nonadaptive method (∘·∘, white circles, dotted line) with $N$ ranging from 50 to 1000 and the adaptive method (•—•, black circles, solid line) with $H/h$ ranging from 2 to 20. For the adaptive method, $N$ corresponds to the number of particles that would be needed in a nonadaptive method with particle size $h$.*

information between fine and coarse interparticle-spacing domains do not deteriorate the expected accuracy at the finest level of refinement.

**5.2. Inviscid 2D vortical flows.** In order to illustrate the capabilities of the presented methods in a multidimensional setting we consider the inviscid evolution of an elliptical vortex of compact support in an unbounded domain. This particular case has already been investigated in detail using a nonadaptive vortex method in [12]. The governing equations for this flow are the 2D incompressible Euler equations in a vorticity-velocity formulation:

$$\begin{aligned} \frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega) &= 0, \\ \nabla \times \mathbf{u} &= \omega, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{5.14}$$

Using the compact vorticity profile (see Figure 11)

$$\omega^I(z) = 20(1 - \exp(-(2.56085/z)\exp(1/(z-1)))),$$

the initial condition is constructed as

$$\omega_o(x,y) = \omega^I(\sqrt{(x/0.8)^2 + (y/1.6)^2}), \tag{5.15}$$

and the velocity satisfies

$$\lim_{|\mathbf{x}|\to\infty} \mathbf{u}(\mathbf{x}) = 0.$$

As an integral diagnostic measure we compute the error in the evolution of the enstrophy as

$$\text{rel. Enstrophy Error} = \mathcal{E}(t)/\mathcal{E}(0),$$

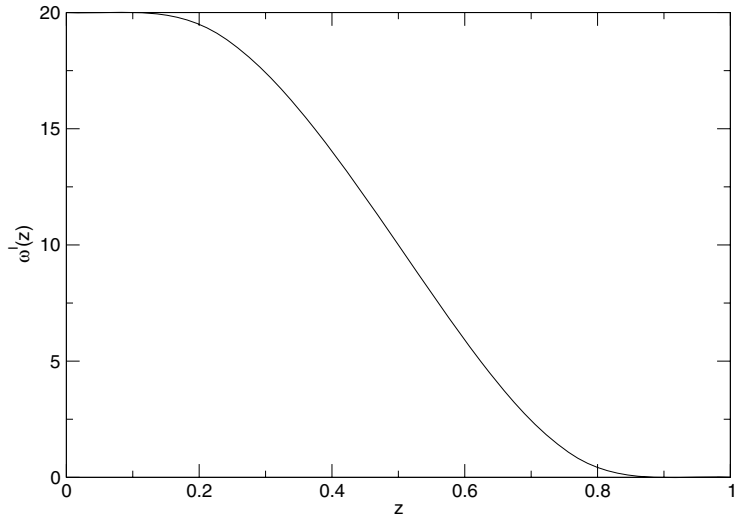


FIG. 11. *Compactly supported vorticity profile used for the initial condition of the vortical flow.*

where the enstrophy is defined as

$$\mathcal{E}(t) \equiv \iint \omega(\mathbf{x},t)^2 \, d\mathbf{x}. \tag{5.16}$$

Additionally, we compare the evolution of the palinstrophy of our methods to the palinstrophy of a high-resolution reference calculation. The palinstrophy is defined as

$$\mathcal{P}(t) = \int\int |\nabla\omega(\mathbf{x},t)|^2 \, d\mathbf{x}.$$

**5.2.1. Adaptive global mapping-based method.** The discretized equations for the above described example are

$$\begin{aligned} \frac{d\hat{\mathbf{x}}_j}{dt} &= \tilde{\mathbf{u}}_j, \\ \frac{d\hat{\Gamma}_j}{dt} &= 0, \\ \frac{d\boldsymbol{\chi}_i}{dt} &= \boldsymbol{\mathcal{U}}(\boldsymbol{\xi}_i), \end{aligned} \tag{5.17}$$

with

$$\tilde{\mathbf{u}} = \underline{\boldsymbol{\Phi}}(\hat{\mathbf{u}} - \boldsymbol{\mathcal{U}}). \tag{5.18}$$

The velocities $\hat{\mathbf{u}}$ are computed in physical space using the fast multipole method (FMM) [2], and we used the same time-stepping technique as for the 1D examples. The map adaptation velocity $\boldsymbol{\mathcal{U}}$ has been computed using the MMPDE (3.22), and as the monitor function we chose

$$M(\hat{\mathbf{x}},t) = \sqrt{1 + \alpha|\mathcal{B}\hat{\omega}|^2}, \tag{5.19}$$

where $\mathcal{B}$ is a B-Spline-based high-pass filter which detects small structures in the vorticity field. No map smoothing has been used, and the particles are remeshed

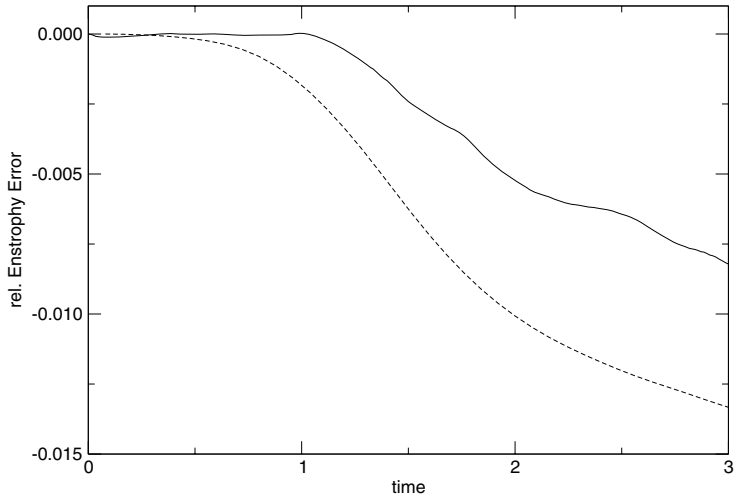


FIG. 12. *Evolution of relative enstrophy error for a nonadaptive method ( – – , dashed line) and the adaptive global mappings-based method ( —— , solid line).*

every time step in the reference space. Figure 12 depicts the evolution of the enstrophy error from $t = 0$ to $t = 3$ for a nonadaptive calculation and for the adaptive global mappings-based method. Particle sizes have been chosen such that they discretize the initial condition with approximately the same amount of particles. It is evident that at early times the adaptive method is able to resolve scales that the nonadaptive method cannot. Around $t = 1.2$—as the amount of small scales in the vorticity field increases—also the adaptive method fails to resolve all scales and begins to show an enstrophy decay similar to the nonadaptive method. Due to the dynamic adaptation of the smallest numerical length scale the adaptive method is also able to follow the evolution of the vorticity gradients more closely than the nonadaptive method. This is illustrated in Figure 13, which shows the evolution of palinstrophy. Figure 14 displays the numbers of particles used in this calculation. As expected, the adaptive method is more efficient in representing the scales in the flow. The number of particles for the adaptive method levels off around 16,000 as the map adapts and expands to account for the growing support of vorticity.

Figure 15 depicts the adaptive mapping at $t = 1.5$ and illustrates how the MMPDE (3.22) is able to attain high compression ratios while maintaining a regular map. It is a well-known fact that vortex methods are often capable of representing correct flow physics even in underresolved cases due to their inherent filtering properties [5]. The characteristics of the adaptive method regarding this property are a subject of ongoing investigations (see Figure 16).

**5.2.2. AMR-based method.** The discretized evolution equations for this case are

$$\begin{aligned} \frac{d\mathbf{x}_j}{dt} &= \mathbf{u}_j, \\ \frac{d\Gamma_j}{dt} &= 0. \end{aligned} \tag{5.20}$$

To simplify the treatment of particles of different sizes in the overlapping zones in the FMM, we have chosen to compute velocities based on the circulation of coarse-size

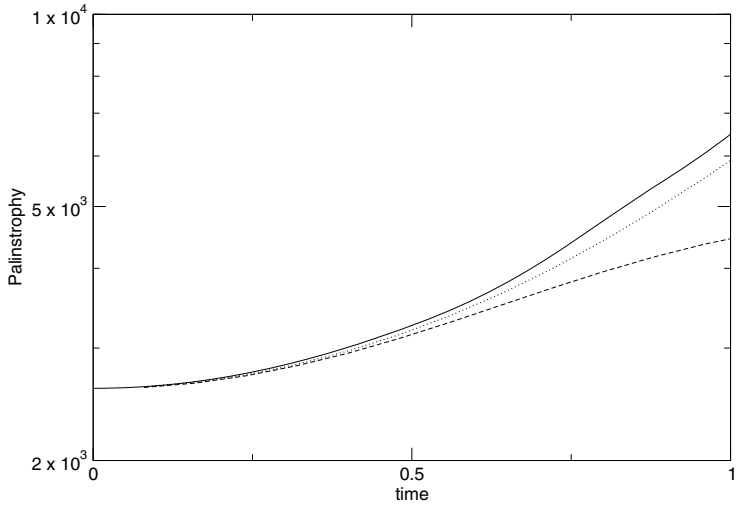


FIG. 13. *Evolution of palinstrophy for a nonadaptive method ( – – , dashed line), the adaptive global mappings-based method ( —— , solid line), and a high-resolution reference calculation ( ···· , dotted line).*

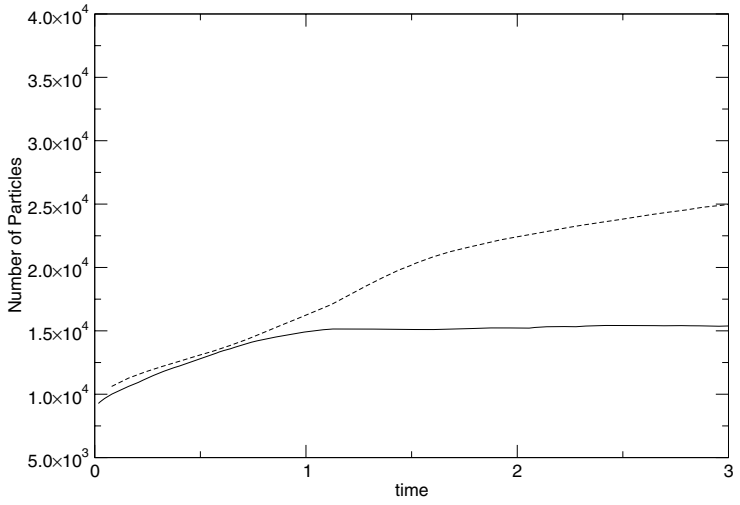


FIG. 14. *Number of particles used in the course of the simulation for the nonadaptive method ( – – , dashed line) and the adaptive global mappings-based method ( —— , solid line); the high-resolution reference calculation used an average of* 670,000 *particles (not shown).*

particles everywhere. The velocities at the locations of fine-size particles have been interpolated using the $M_4'$ interpolation formula.

The coarse-fine domains have been generated as follows: the same indicator for small scales as for the adaptive global mappings-based method is applied to the vorticity field as

$$\mathcal{I}_j(t) = |(\mathcal{B}\omega)(\mathbf{x}_j, t)|, \tag{5.21}$$

and 5% of the particles with the highest values of $\mathcal{I}_j(t)$ are selected. These selected particles are fed into a clustering algorithm [7]. This algorithm, designed for AMR

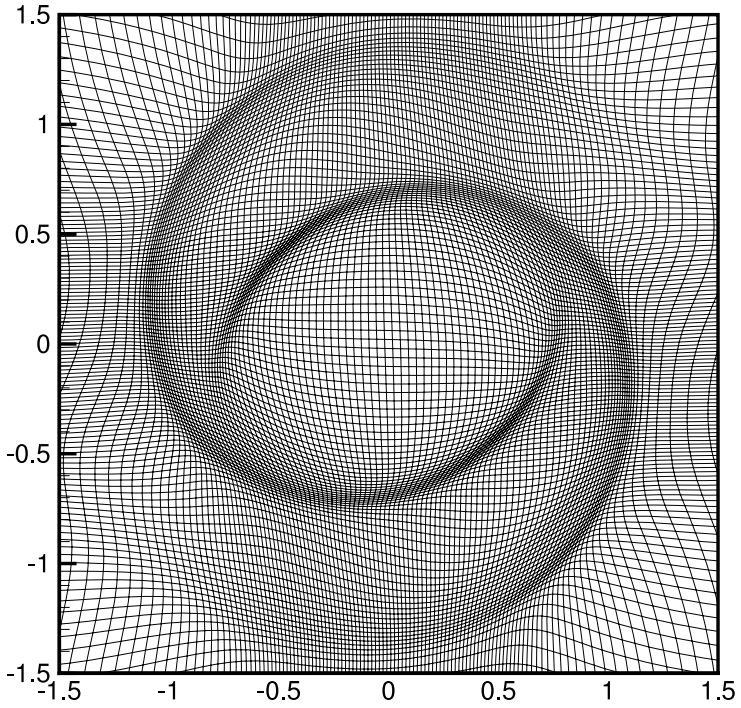


FIG. 15. *Adaptive mapping at $t = 1.5$; adaptation to small scale structures using a high-pass filter in the monitor function.*

finite difference methods, generates a set of rectangular domains which encompass the selected coarse-size particles and where fine-size particles are spawned. In this example we have used one level of refinement with a refinement factor $H/h$ of 4. As the domains are selected anew in every time step, it is eventually necessary to generate fine particles in previously unrefined areas of the computational domain. This is accomplished by interpolation of coarse-size particles using the $M_4'$-kernel.

In Figure 17 we observe the same decay characteristics for the enstrophy decay of the AMR-based method as for the method based on adaptive global mappings; the adaptive method is able to maintain the initial enstrophy level for longer times until the solution becomes underresolved. Figure 18 suggests that the adaptive method is able to capture high vorticity gradients slightly longer than the nonadaptive method. As we are only using one level of refinement this behavior is expected due to the rapid development of small scales in this inviscid problem. The coarse particle size for the AMR-based method has been chosen to be $H = 0.0372$, such that it yields approximately the same number of particles as the nonadaptive method with $H = 0.02$ (see Figure 19). However, more particles are necessary for the qualitatively same result if compared to the adaptive global mappings-based method. Clearly, the efficiency of the AMR-based method could be significantly improved by allowing multiple levels of refinement, an option we have not yet implemented.

Figure 20 displays the dynamic distribution of domains of refinement generated by the clustering algorithm at $t = 1.5$.

Like the adaptive global mapping-based method the AMR-based method is also capable of capturing the correct dynamics of the ellipse (see Figure 21). The filtering

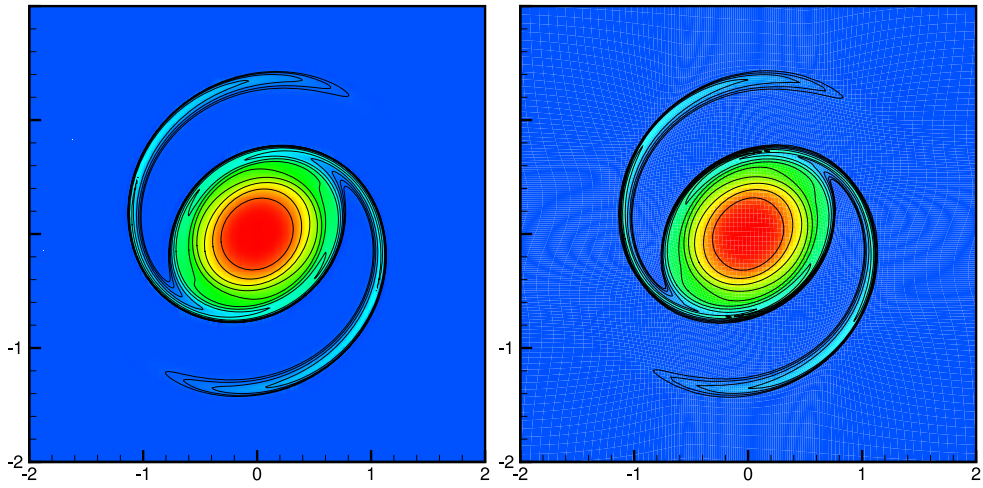


FIG. 16. *Vorticity contours for the high-resolution reference calculation at* $t = 1.5$ *with* $N = 60'800$ *(left) and the corresponding contours for the adaptive global mappings-based method with* $N = 15'100$ *(right).*

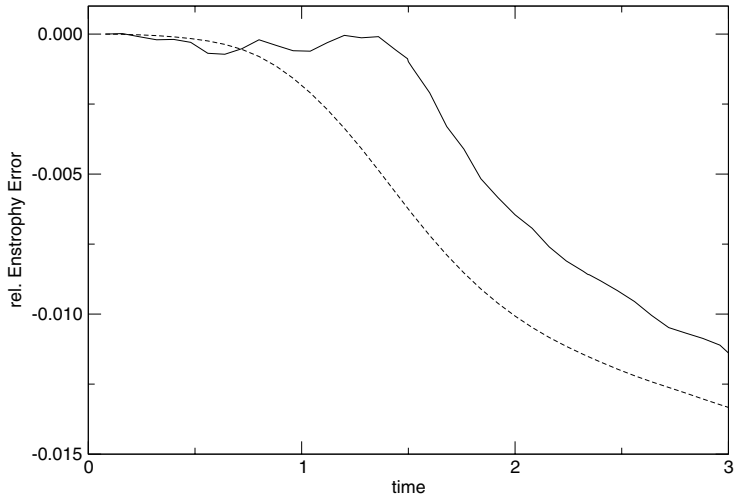


FIG. 17. *Evolution of relative enstrophy error for a nonadaptive method (* – – *, dashed line) and the AMR-based method (* —— *, solid line).*

characteristics are far more obvious for this method, as the filter size is piecewise constant across the computational domain.

### 5.2.3. Comparison of the two methods.

*Generality.* The AMR-based method seems to be applicable to a broader range of problems as areas of refinement can be generated *on demand*, whereas the method based on adaptive global mappings can only adapt transiently and must adapt smoothly to the problem.

*Accuracy.* In terms of accuracy our findings from the 1D cases are applicable to the multidimensional case: The method based on adaptive global mappings can easily attain high particle size ratios while remaining accurate. However, there exists

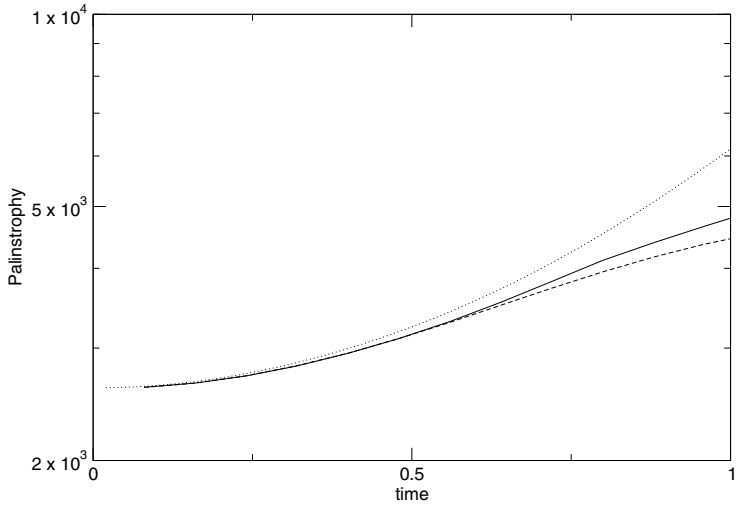


FIG. 18. *Evolution of palinstrophy for a nonadaptive method ( – – , dashed line), the AMR-based method ( —— , solid line), and a high-resolution reference calculation ( ···· , dotted line).*

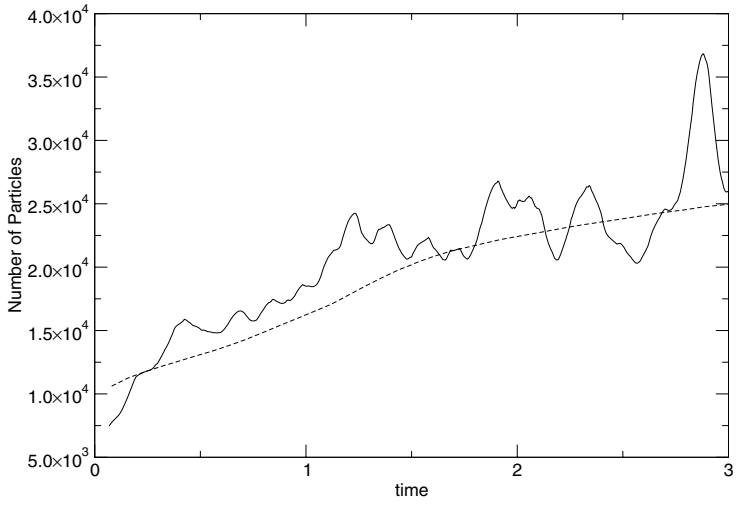


FIG. 19. *Number of particles used in the course of the simulation for the nonadaptive method ( – – , dashed line) and the AMR-based method ( —— , solid line); the high-resolution reference calculation used an average of* 670,000 *particles (not shown).*

an optimal adaptation, which in general is unknown, as too high particle size ratios can lead to increasing error. The AMR-based method can achieve high particle size ratios by employing a sequence of levels of refinement, and our results show that near-optimal error can be achieved. However, as the particle sizes are piecewise constant, it is necessary to prevent aliasing effects by using a stepwise transition from coarse-size to fine-size scales. The same situation is in general the case in grid-based AMR.

*Computational cost.* In our implementations most of the CPU time was spent in the FMM for the evaluation of particle velocities. The computational cost of all algorithmic elements other than the solution of the Poisson equation comprises 12% for the nonadaptive method, 18% for the AMR-based method, and 21% for the adaptive

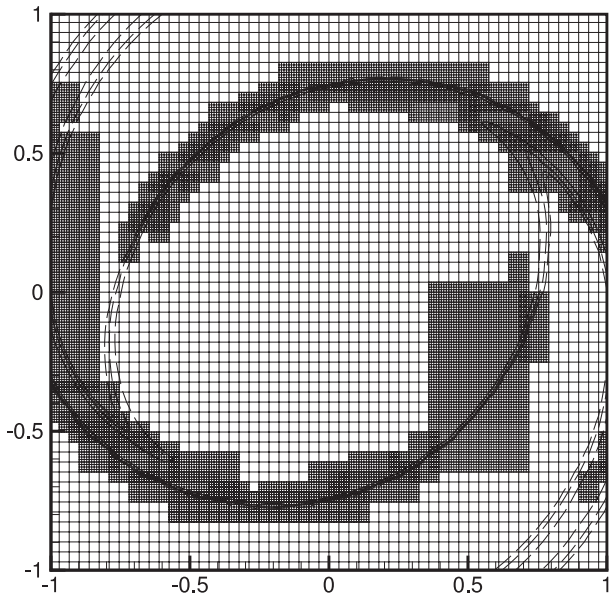


FIG. 20. *Dark (fine grid) areas depict domains of fine-size particles at* $t = 1.5$.

global mappings-based method, and thus represent modest additional cost. Note, however, that the AMR framework makes it possible to envision the use of multilevel grid-based Poisson solvers, leading to methods in the spirit of vortex-in-cell methods and to substantial savings in the velocity evaluations.

*Ease of implementation.* Efficient implementation of the AMR-based method is clearly more challenging than the implementation of an adaptive global mappings-based method, as the latter is still a one-scale particle method in reference space. However, solving the resulting ODEs is more complicated, especially when multistage integrators are used as the map and the particle quantities are coupled.

**6. Conclusion.** We have presented two novel approaches to enhance the adaptivity of Lagrangian particle methods.

In the first approach, we have designed a method enabling r-adaptivity in particle methods by using finite-dimensional mappings from a reference space where all particles have the same core size to a "physical" space where their core sizes vary spatially. We have suggested a simple choice for the map adaptation in the case of 1D convection-dominated problems and the use of MMPDEs for multidimensional problems. Numerical experiments on nonlinear convection in one dimension have shown beneficial effects of such an adaptation, and the application of the method to the 2D incompressible Euler equations showed its extensibility to multidimensions.

In the second approach, an AMR technique inspired by finite difference methods has been proposed and validated. Overlapping of fine- and coarse-resolution domains and appropriate remeshing strategies are the key elements to ensure a consistent exchange of information at each time step between particles at different levels of

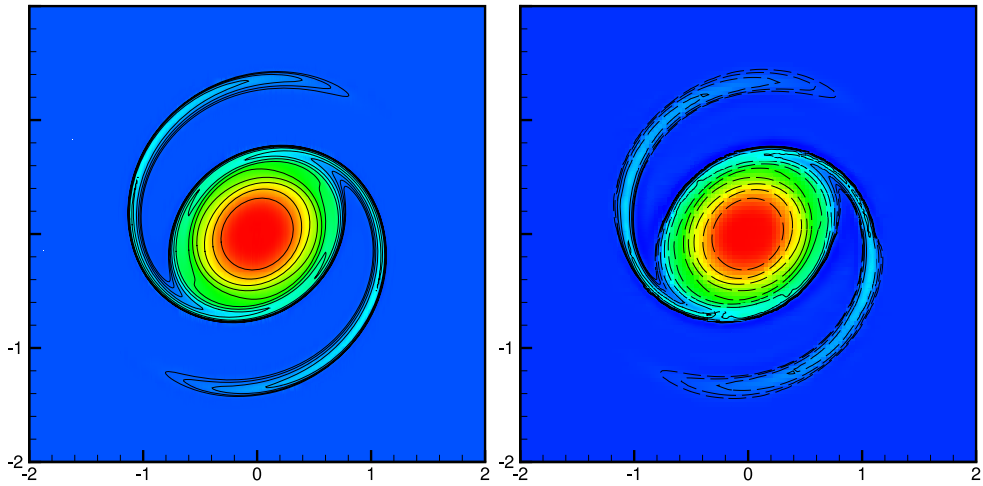


FIG. 21. *Vorticity contours for the high-resolution reference calculation at* $t = 1.5$ *with* $N = 60,800$ *(left) and the corresponding AMR-based contours (* —— *, solid lines correspond to refinement areas,* – – *, dashed lines correspond to coarse areas) with* $N = 20,500$ *(right).*

refinement.

In terms of accuracy, the first approach seems to be optimal, as it allows us to reach levels of refinement that can span several orders of magnitude. The proposed method appears to combine the most attractive features of particle methods and finite-dimensional mappings.

Concerning the second type of methods, their simplicity of implementation make them an appealing tool which can be extended to several dimensions in a straightforward fashion. Applications of that method to gas dynamics with high density gradients is underway. The implementation of multiple levels of refinement for the AMR-based method greatly enhance its efficiency and make it equivalent in accuracy to the first method while remaining more general. Another appealing aspect is the possibility of using efficient grid-based multilevel Poisson solvers.

Possible improvements for both approaches include the development of a posteriori error indicators for the evaluation of the PSE operators and the remeshing process, which then could be used as the monitor function and refinement criterion, respectively.

**Acknowledgments.** We would like to thank Laurent Debreu for his help using the `Agrif`-clustering algorithm and Jens Honore Walther for kindly providing his implementation of the 2D FMM.

## REFERENCES

[1] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484–512.

[2] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686.

[3] H. D. CENICEROS AND T. Y. HOU, *An efficient dynamically adaptive mesh for potentially singular solutions*, J. Comput. Phys., 172 (2001), pp. 609–639.

[4] G.-H. COTTET, *Artificial viscosity models for vortex and particle methods*, J. Comput. Phys., 127 (1996), pp. 299–308.

[5] G.-H. Cottet and P. D. Koumoutsakos, *Vortex Methods. Theory and Practice*, Cambridge University Press, Cambridge, UK, 2000.
[6] G.-H. Cottet, P. D. Koumoutsakos, and M. L. Ould Salihi, *Vortex methods with spatially varying cores*, J. Comput. Phys., 162 (2000), pp. 164–185.
[7] L. Debreu and C. Vouland, *Agrif: Adaptive Grid Refinement in Fortran*, http://www-lmc.imag.fr/IDOPT/AGRIF.
[8] P. Degond and S. Mas-Gallic, *The weighted particle method for convection-diffusion equations.* I. *The case of an isotropic viscosity*, Math. Comp., 53 (1989), pp. 485–507.
[9] P. Degond and S. Mas-Gallic, *The weighted particle method for convection-diffusion equations.* II. *The anisotropic case*, Math. Comp., 53 (1989), pp. 509–525.
[10] T. Y. Hou, *Convergence of a variable blob vortex method for the Euler and Navier–Stokes equations*, SIAM J. Numer. Anal., 27 (1990), pp. 1387–1404.
[11] W. Huang, Y. Ren, and R. D. Russell, *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM J. Numer. Anal., 31 (1994), pp. 709–730.
[12] P. Koumoutsakos, *Inviscid axisymmetrization of an elliptical vortex*, J. Comput. Phys., 138 (1997), pp. 821–857.
[13] K. Lindsay and R. Krasny, *A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow*, J. Comput. Phys., 172 (2001), pp. 879–907.
[14] J. E. Martin and E. Meiburg, *Numerical investigation of three-dimensional evolving jets subject to axisymmetric and azimuthal perturbation*, J. Fluid. Mech., 230 (1991), pp. 271–271.
[15] K. Miller and R. N. Miller, *Moving finite elements.* I, SIAM J. Numer. Anal., 18 (1981), pp. 1019–1032.
[16] J. J. Monaghan, *Extrapolating B-splines for interpolation*, J. Comput. Phys., 60 (1985), pp. 253–262.
[17] P. Ploumhans and G. S. Winckelmans, *Erratum: Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry*, J. Comput. Phys., 170 (2001), pp. 449–456.
[18] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* II, J. Comput. Phys., 83 (1989), pp. 32–78.
[19] H. Viviand, *Formes conservatives des équations de la dynamique des gaz*, Rech. Aérospat., 1 (1974), pp. 65–66.