A Practical Guide to Deterministic Particle Methods

Alina Chertock*

Abstract

The past several decades have seen significant development in the design and numerical analysis of particle methods for approximating solutions of PDEs. In these methods, a numerical solution is sought as a linear combination of Dirac delta-functions located at certain points. The locations and coefficients (weights) of the delta-functions are first chosen to accurately approximate the initial data and then are evolved in time according to the system of ODEs obtained from a weak formulation of the considered problem. The main advantage of the particle methods is their low numerical diffusion that allows them to capture a variety of nonlinear waves with a high resolution. Even though the most "natural" application of the particle methods is linear transport equations, over the years, the range of these methods has been extended for approximating solutions of convection-diffusion and dispersive equations and general nonlinear problems.

In this paper, we provide a mathematical introduction to deterministic particle methods and review different aspects of their practical implementation such as recovering an approximate solution from its particle distribution and an investigation of various particle redistribution algorithms.

1 Introduction

In recent years, particle methods have become a useful tool for approximating solutions of PDEs and been successfully used to treat a broad class of problems arising in astrophysics, plasma physics, solid state physics, medical physics, and fluid dynamics; see, e.g., [29,31,58,69–71,73,82,83] and references therein. In these methods, the solution is sought as a linear combination of Dirac distributions whose positions and coefficients represent locations and weights of the particles, respectively. The solution is then found by following the time evolution of the locations and the weights of the particles according to a system of ODEs, obtained by considering a weak formulation of the problem. In order to recover point values of the the computed solution at some time $t > 0$, the particle solution needs to be regularized, and hence the performance of the particle method depends on the quality of the regularization procedures, allowing the recovery of the approximate solution from its particle distribution. A commonly used regularization of a particle solution performed by taking a convolution with a so-called cut-off function, which plays the role of a smooth approximation to the δ-function and after a proper scaling takes into account the tightness of the particle discretization. When a particle method is applied to problems with nonsmooth data, the reconstruction procedure becomes the most challenging part of the overall algorithm — it works perfectly fine for smooth functions, but may break down when applied to nonsmooth (discontinuous) solutions.

Mesh-free particle methods have many advantages compared to Eulerian (finite-difference, finite-volume, finite-element etc.) methods. The amount of numerical viscosity introduced by most nonoscillatory Eulerian discretizations of the convective terms may seriously degrade the accuracy of a computational method especially if a coarse grid is forced to be used. Lagrangian-type methods, on the other hand, can ameliorate most of the problems posed by the presence of numerical viscosity since particles provide a non-dissipative approximation of the convection. Furthermore, in some scientific

*Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA chertock@math.ncsu.edu

applications, such as kinetic theory, for instance, FD schemes cannot be applied to a realistic case, because of the dimensionality of the problem [33], while in particle schemes, the particles are concentrated in the relevant region of the phase space, optimizing the memory storage of the computer. As mesh-free, particle methods are also very flexible, and, therefore beneficial, when problems with very complicated geometries and/or moving boundaries are considered.

Particle methods have been used for a long time to give a numerical solution of purely convective problems, such as the incompressible Euler equation in fluid mechanics [70, 82, 83] or the Vlasov equation in plasma physics [61]. Over the years, the range of particle methods has been extended to treat other type of equations including convection-diffusion, dispersive and others and we refer the reader to several books and survey monographs, where a large variety of particle-type methods are being reviewed, see,e.g., [40, 84, 95] and references therein. Particle method approximations of hyperbolic PDEs with oscillatory solutions were studied in [54]. A detailed description of particle methods with an emphasis on vortex methods and smooth particle hydrodynamics (SPH) and their applications can also be found in [40, 94].

It is generally possible to divide the particle methods for convection-diffusion equations into two classes: stochastic and deterministic ones. The most widely used treatment of diffusion terms, the random vortex method, was developed in [30]. There, diffusion was introduced by adding a Wiener process to the motion of each vortex. Numerous works followed that pioneering paper and properties of the random vortex method have been extensively studied in the literature (for a comprehensive list we refer the reader to [101] and [40]). Several deterministic methods have been explored for treating the diffusion terms in particle schemes. Among them is the so-called weighted particle method [42, 43, 49, 90, 91]), in which the convective part of the equation is modeled by the convection of the particles, while the diffusion part of the equation is taken into account by changing the particle weights. Another example is the diffusion-velocity method, which is based on defining the convective field associated with the heat operator which then allowed the particles to convect in a standard way ([50, 78, 79, 86]), and others ([34, 52, 53, 59, 104–106]). In this case, the PDE is rewritten as if it was an advection equation with a speed that depends on the solution and its derivatives. The solution is then obtained by implementing the particle method with the only difference being that the point values of the computed solution should be recovered from the particle distribution at every time step during the time integration (unlike linear advection problems, in which the solution is recovered only at the final time). The diffusion-velocity particle method has been also applied to linear and nonlinear dispersive equations as well as used for direct simulations of solitary waves interactions, see, e.g., [12, 16, 24–28, 47, 48, 92].

One must be aware, however, that the self-adaptivity of the particle positions to the local flow map comes at the expense of the regularity of the particle distribution: inter-particle distances may change in time, and just as particles may cluster in the immediate region of the discontinuity they may spread too far from each other near nonsmooth fronts. This may lead not only to a poor resolution of the computed solution, but also to an extremely low efficiency of the methods. The latter is related to the fact that the time step for the ODE solver used to evolve the particle system in time depends, in general, on the distance between the particles. The success of various particle i methods relies thus not only upon accurate reconstruction procedures used to recover point values of the numerical solution from its particle distribution, but also upon accurate and efficient redistribution algorithms, which will ensure that different regions in the computational domain are adequately resolved. A large variety of remeshing techniques were proposed in the literature over the last decades including both global interpolation-type methods, see, e.g, [7–10, 14, 37, 37, 38, 40, 40, 41, 44, 44, 45, 65, 74, 76, 80, 88, 93, 96–98, 100, 100, 102, 108, 110, 111] and local particle merger algorithms, see, e.g., [23, 81, 109]. it is also well-known that particle methods encounter difficulties in the accurate treatment of boundary conditions, while their adaptivity is often associated with severe particle distortion that may introduce spurious scales. Recent research efforts that attempt to address these issues are outlined in [75].

There are many applications, for which a hybridization of the Eulerian and Lagrangian approaches may be beneficial or even crucial for achieving high resolution of the computed solution since particle methods have their own applicability limitations if the considered problems involve additional terms besides linear convection (e.g. collision terms, diffusion, or dispersion, and/or nonlinear terms). Hybrid methods involve combination of mesh-based schemes and particle methods in an effort to utilize the specific advantages of each part of the hybrid method in the right place, see, e.g., [15, 17–20, 22, 35, 36, 46, 70, 97]. It should also be noted, that since in many problems where some sharp interfaces are needed to be captured, the use of particle methods may be critical for obtaining a high resolution of the computed solution due to the low dissipativeness nature of these methods, see also [55, 56, 72, 77].

The purpose of this manuscript is to provide a mathematical review of deterministic particle methods for advection equations as well as to discuss various aspects related to the practical implementation of these methods. The paper is organized as follows: We start, in Section 2, with a description of deterministic particle methods in the context of linear transport equations and provide a review of major analytical results. We then discuss, in Section 3, various remeshing techniques that ensure consistent, efficient, and accurate simulations by particle methods. We conclude, in Section 4, with particle approximations for derivative operators, which are presented in the context of convection-diffusion equations.

2 Description of the Particle Method

In this section, we describe the derivation of the particle method using the example of linear transport equation, here written in the divergence form:

$$u_t + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{a}u) + a_0 u = S, \qquad \boldsymbol{x} \in \mathbb{R}^d, \qquad t > 0, \tag{2.1}$$

and considered subject to initial data

$$u_0(\boldsymbol{x}) := u(\boldsymbol{x}, 0), \tag{2.2}$$

where, u is an unknown function of a time variable t and d-spatial variables $\boldsymbol{x} = (x_1, \ldots, x_d)^T$, the velocity vector $\boldsymbol{a} = (a_1(\boldsymbol{x}, t), \ldots, a_d(\boldsymbol{x}, t))^T$, the coefficient $a_0(\boldsymbol{x}, t)$, and the source/sink term $S(\boldsymbol{x}, t)$ are given functions.

As it was mentioned above, the main idea of the particle methods is to seek a solution of a PDE as a linear combination of Dirac distributions,

$$u^N(\boldsymbol{x}, t) = \sum_{i=1}^{N} w_i(t)\, \delta(\boldsymbol{x} - \boldsymbol{x}_i(t)),$$

for some set $(\boldsymbol{x}_i(t), w_i(t))$ of points $\boldsymbol{x}_i(t) \in \mathbb{R}^d$ and coefficients $w_i(t) \in \mathbb{R}$ that are chosen at time $t = 0$ to accurately approximate the initial data and then evolved in time according to the system of ODEs obtained from a weak formulation of the underlying PDE. Such solutions are called particle solutions and the δ-functions in the above formula are called particles. The coefficients $w_i(t)$ are called particle weights, since they represent the amount of the physical quantity u, carried by the ith particle, which is located at $\boldsymbol{x}_i(t)$ at time t, and N is the total number of particles. It should be emphasized that the introduced particles are mathematical objects rather than (physical) particles of a certain material. An important step in implementation of particle methods is the recovery of point values of the computed solution from its particle distribution. In what follow, we describe each one of the aforementioned steps, i.e., initialization, evolution and reconstruction.

2.1 Particle Approximation of the Initial Data

The first step in the derivation of the particle method consists of approximating the initial data (2.2) by a linear combination of Dirac distributions:

$$u_0^N(\boldsymbol{x}) = \sum_{i=1}^{N} w_i(0)\,\delta(\boldsymbol{x} - \boldsymbol{x}_i(0)), \tag{2.3}$$

where $w_i(0)$ are given coefficients and $\boldsymbol{x}_i(0)$ are the initial locations of the δ-functions. This can be done, for instance, in the sense of measures. Namely, for any test function $\phi \in C_0^0(\Omega)$, the inner product $(u_0(\cdot), \phi(\cdot))$ should be approximated by

$$\int\limits_{\Omega} u_0(\boldsymbol{x})\phi(\boldsymbol{x})\,d\boldsymbol{x} \approx (u_0^N(\cdot), \phi(\cdot)) = \sum_{i=1}^{N} w_i(0)\phi(\boldsymbol{x}). \tag{2.4}$$

Based on (2.4), we observe that that determining the initial weights, $w_i(0)$, is exactly equivalent to solving a standard numerical quadrature problem. One way of solving this problem is to first divide the computational domain Ω into a set of N nonoverlapping subdomains Ω_i: $\bigcup\limits_{i=1}^{N} \Omega_i = \Omega,\ \Omega_i \bigcap \Omega_l = \emptyset,\ \forall i \neq l$. Then the location of the ith particle, $\boldsymbol{x}_i(0)$, is set at the center of mass of Ω_i and the entire mass of u_0 in Ω_i is "placed" into the ith particle so that its initial weight is

$$w_i(0) := \int\limits_{\Omega_i} u_0(\boldsymbol{x})\,d\boldsymbol{x}. \tag{2.5}$$

For instance, one can take

$$w_i(0) = |\Omega_i|\,u_0(\boldsymbol{x}_i(0)),$$

which will correspond to the midpoint rule for (2.5). In general, one can build a sequence of basis functions that will aid in solving the numerical quadrature problem given by (2.4), see, e.g., [26, 27]. One may also prove (see, e.g., [26, 40, 103]) that u_0^N converges weakly to u_0 as $N \to \infty$, i.e., if

$$\max_i |\Omega_i| \to 0 \quad \text{as} \quad N \to \infty, \tag{2.6}$$

then for all $\phi \in C_0^0(\Omega)$

$$\lim_{N\to\infty} \int\limits_{\Omega} (u_0^N(\boldsymbol{x}) - u_0(\boldsymbol{x}))\phi(\boldsymbol{x})\,d\boldsymbol{x} = 0.$$

2.2 Time Evolution of Particles

As it was mentioned above, particle methods are obtained by considering a weak formulation of the underlying problem and therefore we start by defining a weak solution of (2.1), (2.2). Following [103], we denote by $\mathfrak{M}(\Omega)$ the space of measures defined on $\Omega \subset \mathbb{R}^d$, that is, the space dual to the space $C_0^0(\Omega)$ of continuous functions $\Omega \to \mathbb{R}$ with compact support. A weak solution is then defined as follows.

Definition 2.1 *A function $u \in \mathfrak{M}(\mathbb{R}^d \times [0,T))$ is called a* **weak solution** *of the Cauchy problem (2.1), (2.2) if*

$$\begin{aligned} -\int_{\mathbb{R}^d} u_0(\mathbf{x})\varphi(\boldsymbol{x},0)\,d\boldsymbol{x} - \int_0^T\int_{\mathbb{R}^d} u(\boldsymbol{x},t)\left[\varphi_t(\boldsymbol{x},t) + \boldsymbol{a}(\boldsymbol{x},t)\cdot\nabla\varphi(\boldsymbol{x},t)\right]\,d\boldsymbol{x}\,dt \\ + \int_0^T\int_{\mathbb{R}^d} a_0(\boldsymbol{x},t)u(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,d\boldsymbol{x}\,dt = \int_0^T\int_{\mathbb{R}^d} S(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,d\boldsymbol{x}\,dt \end{aligned} \tag{2.7}$$

holds for any test function $\varphi \in C_0^1(\mathbb{R}^d \times [0,T))$.

Note that this definition makes sense if $u_0 \in \mathfrak{M}(\mathbb{R}^d)$ and $S \in \mathfrak{M}(\mathbb{R}^d \times [0,T))$.

Equipped with the definition of a weak solution, we now prove the following proposition, which holds in the homogeneous case of $S \equiv 0$.

Proposition 2.1 *Let*

$$u_t + \boldsymbol{\nabla}_{\boldsymbol{x}}\cdot(\boldsymbol{a}u) + a_0(\boldsymbol{x},t)u = 0, \tag{2.8}$$

be a homogeneous linear transport equation considered subject to the initial data (2.3), where $w_i(0)$ are given coefficients and $\boldsymbol{x}_i(0)$ are the initial locations of the δ-functions. Assume that $a_0, a_1, \ldots, a_d \in C(\mathbb{R}^d \times [0,T])$. Then a weak solution of (2.8), (2.3) is given by

$$u^N(\boldsymbol{x},t) = \sum_{i=1}^N w_i(t)\,\delta(\boldsymbol{x} - \boldsymbol{x}_i(t)), \tag{2.9}$$

where

$$\begin{aligned} &\frac{d\boldsymbol{x}_i(t)}{dt} = \boldsymbol{a}(\boldsymbol{x}_i(t),t), \\ &\frac{dw_i(t)}{dt} + a_0(\boldsymbol{x}_i(t),t)\,w_i(t) = 0, \quad i = 1,\ldots,N. \end{aligned} \tag{2.10}$$

Proof: Let $\varphi \in C_0^1(\mathbb{R}^d \times [0,T))$. Substituting (2.9) into the weak formulation (2.7) and changing the order of summation and integration yields:

$$\begin{aligned} -\sum_{i=0}^N w_i(0)\varphi(\boldsymbol{x}_i(0),0) - \sum_{i=0}^N \int_0^T w_i(t)\left[\varphi_t(\boldsymbol{x}_i(t),t) + \boldsymbol{a}(\boldsymbol{x}_i(t),t)\cdot\nabla\varphi(\boldsymbol{x}_i(t),t)\right]\,dt \\ + \sum_{i=0}^N \int_0^T w_i(t)a_0(\boldsymbol{x}_i(t),t)\varphi(\boldsymbol{x}_i(t),t)\,dt = 0. \end{aligned} \tag{2.11}$$

We now add and subtract $\displaystyle\sum_{i=0}^N \int_0^T w_i(t)\,\frac{d\boldsymbol{x}_i(t)}{dt}\cdot\nabla\varphi(\boldsymbol{x}_i(t),t)\,dt$ in the last equation, use the fact that the time derivative of φ along the curve $\boldsymbol{x}_i = \boldsymbol{x}_i(t)$ is

$$\frac{d\varphi(\boldsymbol{x}_i(t),t)}{dt} = \varphi_t(\boldsymbol{x}_i(t),t) + \frac{d\boldsymbol{x}_i(t)}{dt}\cdot\nabla\varphi(\boldsymbol{x}_i(t),t),$$

and rewrite equation (2.11) as follows:

$$
\begin{aligned}
&-\sum_{i=0}^{N} w_i(0)\varphi(\boldsymbol{x}_i(0),0)-\sum_{i=0}^{N}\int\limits_0^T w_i(t)\frac{d\varphi(\boldsymbol{x}_i(t),t)}{dt}\,dt\\
&+\sum_{i=0}^{N}\int\limits_0^T w_i(t)\left[\frac{d\boldsymbol{x}_i(t)}{dt}-\boldsymbol{a}(\boldsymbol{x}_i(t),t)\right]\cdot\nabla\varphi(\boldsymbol{x}_i(t),t)\,dt+\sum_{i=0}^{N}\int\limits_0^T w_i(t)a_0(\boldsymbol{x}_i(t),t)\varphi(\boldsymbol{x}_i(t),t)\,dt=0.
\end{aligned}
\tag{2.12}
$$

Integrating by parts the second term in the first row in (2.12) and rearranging other terms, we finally obtain:

$$
\begin{aligned}
&\sum_{i=0}^{N}\int\limits_0^T w_i(t)\left[\frac{d\boldsymbol{x}_i(t)}{dt}-\boldsymbol{u}(\boldsymbol{x}_i(t),t)\right]\cdot\nabla\varphi(\boldsymbol{x}_i(t),t)\,dt\\
&\qquad+\sum_{i=0}^{N}\int\limits_0^T\left[\frac{dw_i(t)}{dt}+w_i(t)a_0(\boldsymbol{x}_i(t),t)\right]\varphi(\boldsymbol{x}_i(t),t)\,dt=0.
\end{aligned}
\tag{2.13}
$$

Since the functions $\boldsymbol{x}_i(t)$ and $w_i(t)$ satisfy the system (2.10), the last equation holds for any φ implying that u^N defined by (2.9), (2.10) is a weak solution of (2.8), (2.3). This completes the proof. ■

In the general case of $S\neq 0$, a particle solution of (2.1), (2.2) is obtained by solving the initial-value problem (2.1), (2.3), whose solution $u^N(\boldsymbol{x},t)$ is still given by (2.9), but the locations and weights of the particles satisfy a different system of ODEs (compare with the system (2.10)):

$$
\begin{aligned}
&\frac{d\boldsymbol{x}_i(t)}{dt}=\boldsymbol{a}(\boldsymbol{x}_i(t),t),\\
&\frac{dw_i(t)}{dt}+a_0(\boldsymbol{x}_i(t),t)\,w_i(t)=\beta_i(t),
\end{aligned}
\tag{2.14}
$$

where β_i reflects the contribution of the source term S (see, e.g., [32,103]). To evaluate β_i, we consider a particle approximation of S, given by

$$
S(\boldsymbol{x},t)\approx S^N(\boldsymbol{x},t):=\sum_{i=1}^{N}\beta_i(t)\,\delta(\boldsymbol{x}-\boldsymbol{x}_i(t)),
$$

where

$$
\beta_i(t)=\int\limits_{\Omega_i(t)} S(\boldsymbol{x},t)\,d\boldsymbol{x}\approx S(\boldsymbol{x}_i(t),t)\,|\Omega_i(t)|.
\tag{2.15}
$$

Here, $\Omega_i(t)$ is the subdomain of Ω that includes the ith particle and should satisfy the following two properties for all t (not only at $t=0$):

$$
\Omega=\bigcup_{i=1}^{N}\Omega_i(t),\ \ \Omega_i(t)\bigcap\Omega_l(t)=\emptyset,\ \forall i\neq l,\quad w_i(t)\approx\int\limits_{\Omega_i(t)}u(\boldsymbol{x},t)\,d\boldsymbol{x}.
$$

These requirements are hard to guarantee, but noticing that in order to use (2.15) only the size of $\Omega_i(t)$ is needed, one may apply the Liouville theorem (see, e.g., [3,31]) to obtain the following ODE

$$
\frac{d}{dt}|\Omega_i(t)|=\boldsymbol{\nabla}_{\boldsymbol{x}}\cdot\boldsymbol{a}(\boldsymbol{x}_i(t),t)\,|\Omega_i(t)|,
\tag{2.16}
$$

which has to be solved together with the system (2.14) to complete the construction of the particle method.

Remark 2.1 We note that in the case when the velocity vector field $\boldsymbol{u}$ is divergence-free, that is, when $\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{a} \equiv 0$, the size of the ith subdomain does not change, that is, $|\Omega_i(t)| \equiv |\Omega_i|\ \forall t$. Otherwise, $|\Omega_i(t)|$ changes in time and the coefficients (weights) β_i in (2.15) would depend on t even when the source $S = S(\boldsymbol{x})$ only.

Remark 2.2 It should be observed that in practice, except in very special cases, the functions $\boldsymbol{x}_i(t)$ and $w_i(t)$ have to be determined numerically. In fact, the ODE system (2.10) can be solved by means of a classical numerical method using one's favorite ODE solver.

The above discussion leads to the following algorithm for obtaining a particle solution of (2.1), (2.2):

- `Divide the computational domain` Ω `into a set of` N `nonoverlapping subdomains` Ω_i: $\bigcup_{i=1}^{N} \Omega_i = \Omega,\ \ \Omega_i \bigcap \Omega_l = \emptyset,\ \forall i \neq l\,.$
- `Replace the the initial datum` u_0 `in` (2.2) `by its particle approximation` u_0^N, (2.3), `for some set` $(\boldsymbol{x}_i(0), w_i(0))$ `of points` $\boldsymbol{x}_i(0)$ `in the computational domain` Ω `and weights` $w_i(0)$ `computed according to (2.5).`
- `If` $S \equiv 0$ `in the right-hand side of equation (2.1), consider the system (2.8), (2.3) and obtain its particle solution` (2.9) `by numerically integrating the system of ODEs (2.10).`
- `If` $S \neq 0$, `replace` $S(\boldsymbol{x}, 0)$ `with its particle approximation to compute the coefficients` $\beta_i(0)$ `according to` (2.15) `with` $t = 0$. `Consider the system (2.1), (2.3) and obtain its particle solution` (2.9) `by numerically solving the system of ODEs (2.14)-`(2.16).

Finally, the basic convergence result for particle methods is stated in the following theorem.

Theorem 2.1 *Consider equation (2.8) with the coefficients* $a_i \in L^\infty(0, T; W^{1,\infty}(\Omega)),\ i = 1, \ldots, d$ *and* $a_0 \in L^\infty(\Omega \times (0, T))$. *Let the initial condition* $u_0 \in C^0(\Omega)$ *and its particle approximation is given by (2.3). Then, for all* $\varphi \in C_0^0(\Omega)$

$$\lim_{N\to\infty} \int_\Omega (u(\boldsymbol{x}, t) - u^N(\boldsymbol{x}, t))\varphi(\boldsymbol{x})\, d\boldsymbol{x} = 0 \quad \text{uniformly in } t \in [0, T].$$

Theorem 2.1 guarantees that for sufficiently regular coefficients $\boldsymbol{a}, a_0$ and initial datum w_0, the particle solution will converge to the exact one in the sense of measures. However, the above result is of little use when one is interested in obtaining point values of the computed solution. In this respect, it is more useful to associate the particle solution $u^N(\cdot, t)$ with a continuous function $u_\varepsilon^N(\cdot, t)$, which approximates the solution $u(\cdot, t)$ in a more classical sense. In a next section we provide an example of how such an approximation can be constructed.

2.3 Particle Function Approximations

The regularization of particle solution is usually performed by taking a convolution product with a mollification kernel (or, so-called, cut-off function), $\zeta_\varepsilon(\boldsymbol{x})$, that after a proper scaling takes into account the initial tightness of the particle discretization, namely,

$$u(\boldsymbol{x}, t) \approx u_\varepsilon(\boldsymbol{x}, t) := (u(\cdot, t) * \zeta_\varepsilon(\cdot))(\boldsymbol{x}) = \int u(\boldsymbol{y}, t)\zeta_\varepsilon(\boldsymbol{x} - \boldsymbol{y})\, d\boldsymbol{y}, \tag{2.17}$$

where $\zeta_\varepsilon \in C^0(\mathbb{R}^d)\bigcap L^1(\mathbb{R}^d)$ satisfies the following properties

$$\zeta_\varepsilon(\boldsymbol{x}) := \frac{1}{\varepsilon^d}\zeta\left(\frac{\boldsymbol{x}}{\varepsilon}\right), \qquad \int\limits_{\mathbb{R}^d} \zeta(\boldsymbol{x})\,d\boldsymbol{x} = 1, \tag{2.18}$$

and ε denotes a characteristic length of the kernel, see, e.g., [103]. The particle approximation of the regularized solution is then defined as

$$u(\boldsymbol{x},t) \approx u^N_\varepsilon(\boldsymbol{x},t) := (u^N(\cdot,t) * \zeta_\varepsilon(\cdot))(\boldsymbol{x}) = \sum_{i=1}^N w_i(t)\zeta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_i(t)). \tag{2.19}$$

It is clear from the above discussion that the approximation of the computed solution by formula (2.19) is one of the key ingredients of the particle method and hence, the performance of the method depends on the quality of this smoothing procedure. The error introduced by the quadrature (2.19) of the mollified approximation u^N_ε for the function u can be divided into two parts:

$$u - u^N_\varepsilon = (u - u * \zeta_\varepsilon) + (u - u^N) * \zeta_\varepsilon. \tag{2.20}$$

The first term in equation (2.20) denotes the mollification error that can be controlled by appropriately selecting the kernel properties. The second term denotes the quadrature error due to the approximation of the integral in (2.17) on the particle location. The accuracy of the particle method will thus be related to the moments of ζ that are being conserved, and we say that the kernel is of order k when:

$$\begin{cases} \displaystyle\int\limits_{\mathbb{R}^d} \zeta(\boldsymbol{x})\,d\boldsymbol{x} = 1, \\[2ex] \displaystyle\int\limits_{\mathbb{R}^d} \boldsymbol{x}^\alpha\zeta(\boldsymbol{x})\,d\boldsymbol{x} = 0, \quad \text{for all multiindexes } \alpha \text{ such that } 1 \le |\alpha| \le k-1, \\[2ex] \displaystyle\int\limits_{\mathbb{R}^d} |\boldsymbol{x}|^k|\zeta(\boldsymbol{x})|\,d\boldsymbol{x} < \infty. \end{cases} \tag{2.21}$$

Example 2.1 (Gaussian and Generalized Gaussian) A (generalized) Gaussian can be considered as an example of a cut-off function and is obtained by taking the inverse Fourier transform of the function $e^{-|\xi|^{2l}}$, where $l \in \mathbb{N}$ and $\xi \in \mathbb{R}^d$:

$$\zeta_l(\boldsymbol{x}) = C_l \int\limits_{\mathbb{R}^d} e^{i\boldsymbol{x}\cdot\xi}\, e^{-|\xi|^{2l}} d\xi. \tag{2.22}$$

Taking an appropriate normalizing coefficient C_l, one can ensure that the C^∞-functions ζ_l, which rapidly decay at ∞, satisfy the conditions (2.21) with $k = 2l$. Thus, these mollifiers are of order l. In particular, when $l = 1$ we obtain the first-order Gaussian

$$\zeta_1(\boldsymbol{x}) = \frac{1}{\pi^{d/2}} e^{-|\boldsymbol{x}|^2}.$$

When $l = 2$, (2.22) reduces to the second-order super-Gaussian, which in the one-dimensional (1-D) case reads

$$\zeta_2(x) = \frac{1}{\sqrt{\pi}}\left(\frac{3}{2} - x^2\right) e^{-x^2}.$$

Example 2.2 (Compactly Supported Mollifiers) The simplest example of a compactly supported 1-D cut-off function is the quadratic B-spline:

$$\zeta(x) = \begin{cases} \frac{3}{4} - x^2, & |x| \le \frac{1}{2}, \\ \frac{1}{2}\left(\frac{3}{2} - |x|\right)^2, & \frac{1}{2} \le |x| \le \frac{3}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

For this cut-off function $k = 2$, and thus it is only first-order accurate. An advantage of compactly supported cut-off functions is that it is easy to implement the summation in (2.19) in a very efficient way, which is one of the crucial points in practical implementation of particle methods. Higher order compactly supported kernels can be obtained by ensuring that more moments are conserved and can be found in, e.g., [40, 88, 93].

There is an extensive discussion in the literature on the selection of a cut-off function and its relation to the overall accuracy of particle methods, [5–7, 30, 40, 66, 68, 101]. See also [1, 32, 49, 50, 67, 75, 91, 93, 94, 96, 98, 103]. Obviously, the accuracy of the particle method will depend both on the choice of cut-off function (2.18) and on its width ε as it is given by the following theorem [103] (note that similar error estimates can also be obtained for compactly supported cut-off functions, see, e.g., [57, 103]).

Theorem 2.2 *Let* (2.21) *are satisfied for some integer* $k \ge 1$. *Assume that both* ζ *and the coefficients in* (2.1) *are sufficiently smooth, more precisely:* $\zeta \in W^{m,\infty}(\mathbb{R}^d) \bigcap W^{m,1}(\mathbb{R}^d)$ *for some integer* $m > d$ *and* $a_1, \ldots, a_d, a_0 + \nabla \cdot \boldsymbol{a} \in L^\infty(0, T; W^{l,\infty}(\mathbb{R}^d))$, *where* $l = \max(k, m)$. *Then, if the initial datum is also smooth* ($u_0 \in W^{l,p}(\mathbb{R}^d)$), *then there exists a positive constant* $C = C(T)$, *such that for any* $t \in [0, T]$,

$$\|u - u_\varepsilon^N\|_{L^p(\mathbb{R}^d)} \le C \left\{ \varepsilon^k \|u_0\|_{k,p,\mathbb{R}^d} + \left(\frac{h}{\varepsilon}\right)^m \|u_0\|_{m,p,\mathbb{R}^d} \right\}, \tag{2.23}$$

where $h > 0$ *is the size of nonoverlapping* d*-dimensional cubes covering* $\mathbb{R}^d$.

Accuracy. The two terms in the error estimate (2.23) may be balanced by choosing an appropriate size of ε. Intuitively, it is clear that if the smoothing parameter ε is too small in comparison to the minimal distance between particles, the approximate solution defined by (2.19) will vanish away from the ε-neighborhood of the particles and is thus irrelevant. On the other hand, large values of ε will generate unacceptable smoothing errors. Theoretically ε is chosen so that the smoothing error and the discretization error are of the same order and it is common to take $\varepsilon \sim \sqrt{h}$, see, e.g., [20, 21, 67, 103]. It should be noted, however, that it is not clear what the optimal proportionality constant is. That strongly depends on both the smoothness of the flow and the cut-off function and its value at the origin. It has been also observed in various numerical experiments that the overall accuracy of the particle method can deteriorate over long time integrations, see, e.g., the discussion in [7, 65, 67, 75, 96, 98] and in references therein.

Smooth vs. discontinuous solutions. The procedure for point values recovery described above provides an optimal order of accuracy when the captured solution is sufficiently smooth. However, when $\boldsymbol{a}$ and/or u in (2.1) are discontinuous, approximation (2.19) may be inaccurate and even unacceptable, as it has been demonstrated in [18,20,22]. In order to overcome this difficulty, the convolution procedure (2.19) can be, for instance, modified by taking kernels of locally varying width, whose size was selected in a point-wise manner depending on the distance between the particles, [8, 20, 39, 74]. Another approach for recovering point values of the computed solution from its particle distribution is

based on interpreting the particle data as integrals of the approximated solution over some nonoverlapping domains around the particles, see, e.g., [18, 22]. Then, the averaged values of the numerical solution can be obtained dividing the weights $w_i(t)$ by the volumes of the corresponding domains. We also refer the reader to [41], where total variation diminishing (TVD) reconstruction formulae for discontinuous solutions were proposed for particle methods in the context of nonlinear conservation laws.

Efficiency. Besides the accuracy of particle methods, the computational speed is also important. The most time-consuming aspect in particle simulations is accurately evaluating the long-range interactions of particles that appear in the summation formula (2.19). The direct method of computing values of the cut-off function requires $\mathcal{O}(N^2)$ flops, where N is the total number of particles. There is a number of fast summation algorithms known to reduce the computational cost of the particle method to $\mathcal{O}(N\log N)$. Example of such methods include particle-mesh techniques that account for particles in close proximity in terms of grid spacing, see, e.g. [73] and references therein. There are also mesh-free fast summation techniques that are based on the concept of multipole end Taylor expansions, see, e.g., [2, 4, 13, 63, 85]. These methods employ clustering of particles and use expansions of the potentials around the cluster centers with a limited number of terms to calculate their far-field influence onto other particles. These techniques rely on tree data structures to achieve computational efficiency. The tree allows a spatial grouping of the particles, and the interactions of well-separated particles is computed using their center of mass or multipole and Taylor expansions.

3 Remeshing for Particle Distortion

As evidenced by the foregoing discussion, the time evolution of particle positions is dictated by the gradients of the flow field and, as the result, inter-particle distances constantly change. This self-adaptation of particles to the local flow map comes at the expense of the regularity of their distribution since particles may either spread away from each other or cluster in the regions of the large velocity gradients or near discontinuities even if the velocity function $\boldsymbol{a}$ is a smooth function of its arguments. This becomes a critical issue in preserving the accuracy of particle methods and their efficient implementation.

While the lack of particles in certain areas may result in deterioration of the accuracy of the method and poor resolution of the computed solution (appearance of artificial vacuum areas), local particle accumulations may lead to an extremely low efficiency of the method making it almost impractical. The latter is related to the fact that the time step of numerical ODE solvers for (2.10) (or (2.14)) depends on the distance between the particles as the stability condition imposes that no characteristic curves or their trajectories should intersect in finite time. The latter leads to a time-step constraint of the type

$$\Delta t \le C\|\boldsymbol{\nabla a}\|_\infty^{-1}, \tag{3.1}$$

where C depends on the specific numerical solver used. As the result, the success of various particle methods relies not only upon accurate reconstruction procedures used to recover numerical solution from its particle distribution but also upon accurate and efficient particle redistribution algorithms, which will ensure that different regions in the computational domain are adequately resolved.

The redistribution of particles is in general performed through interpolation formulae and consists of (occasional) re-initialization (remeshing) of particle locations onto a new regularized set of particles and recalculating the particle weights on these new locations. The resulting problem of extracting information on a regular grid from a set of scattered points has a long history and we refer the reader to an extensive list of redistribution algorithms, which are based on various techniques ranging from global rezoning using fixed, one-sided and variable size kernels, see, e.g., [7, 40, 44, 65, 74, 93, 96, 98, 100, 108]

and particle-mesh methods, see, e.g., [9,10,14,37,37,38,40,41,44,45,76,80,88,97,100,102,110,111] to multilevel adaptive particle methods, see, e.g., [8,75] and references therein.

3.1 Particle Weights Redistribution

For completeness of the presentation, we provide here several examples that illustrate basic, yet commonly used, global and local remeshing ideas. To this end, we denote by $\{\boldsymbol{x}_i\}_{i=1}^N$ and $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^J$ respectively the old (distorted) and the new (regular) set of particles and we denote by w_i and $\widetilde{w}_j$ the corresponding particle weights at the old and new locations at certain time t. For the sake of simplicity, we also assume that the new particles, $\widetilde{\boldsymbol{x}}_j(t)$, lie at the centers of the nonoverlapping cells, $\{C_j\}_{i=1}^J$, $\bigcup_{j=1}^J C_j = \Omega$, $C_j \bigcap C_l = \emptyset$, $\forall j \neq l$, of size h that cover the computational domain Ω.

Convolution. One of the simplest and natural ways of computing new particle weights $\widetilde{w}_j$ is to reinitialize particle approximation (2.19) at the new nodes by just replacing $\boldsymbol{x}$ by $\widetilde{\boldsymbol{x}}_j(t)$ in the formula, that is,

$$u_\varepsilon^N(\widetilde{\boldsymbol{x}}_j(t),t) = \sum_{i=1}^N w_i(t)\zeta_\varepsilon(\widetilde{\boldsymbol{x}}_j(t) - \boldsymbol{x}_i(t)), \quad j = 1,\ldots,J, \tag{3.2}$$

see, e.g., [7,20,24,25,40,74,93,96]. Using the relation

$$\widetilde{w}_j(t) = \int_{C_j} u(\boldsymbol{x},t)\,d\boldsymbol{x}, \tag{3.3}$$

the new weights can be recomputed by, say, the midpoint rule as $\widetilde{w}_j(t^*) = hu_\varepsilon^N(\widetilde{\boldsymbol{x}}_j(t),t)$, $\forall j = 1,\ldots,J$, and then the particle method is restarted.

Interpolation. Another widespread redistribution procedure is based on a classical interpolation rule:

$$\widetilde{w}_j(t) = \sum_{i=1}^N w_i(t)\,\Phi\left(\frac{|\widetilde{\boldsymbol{x}}_j(t) - \boldsymbol{x}_i(t)|}{h}\right), \quad j = 1,\ldots,J, \tag{3.4}$$

where Φ is an interpolation kernel whose properties determine the type and the quality of the redistribution procedure and is typically designed to conserve a certain number of moments of the particle distribution, see, e.g., [20,21,23,40,73,93,99,107]. This technique (unlike the previously described convolution approach) is in particularly appealing in applications, where, conservation of total amount of u may be crucial for designing a good numerical method. In the context of redistribution of particles, the conservation requirement can be stated as

$$\sum_{j=1}^J \widetilde{w}_j(t) = \sum_{i=1}^N w_i(t). \tag{3.5}$$

There are many choices for $\Phi(r)$ that ensure (3.5) and we consult here [99] to bring two particular examples: a first-order:

$$\Phi(r) = \begin{cases} \frac{1}{8}\left(5 - 2r - \sqrt{-7 + 12r - 4r^2}\right), & \text{if } |r| < 1, \\ \frac{1}{8}\left(3 - 2r - \sqrt{1 + 4r - 4r^2}\right), & \text{if } 1 < |r| < 2, \\ 0, & \text{otherwise} \end{cases}$$

and a second-order kernel:

$$\Phi(r)=\begin{cases}\phi(r), & |r|<1,\\ \frac{1}{6}r^3-\frac{7}{8}r^2+\frac{7}{12}r+\frac{21}{16}-\frac{3}{2}\phi(r-1), & 1<|r|<2,\\ -\frac{1}{12}r^3+\frac{3}{4}r^2-\frac{23}{12}r+\frac{9}{8}+\frac{1}{2}\phi(r-2), & 2<|r|<3,\\ 0, & \text{otherwise},\end{cases}$$

with

$$\begin{aligned}\phi(r)=&\frac{1}{12}r^3-\frac{11}{56}r^2-\frac{11}{42}r+\frac{61}{112}+\\ &\frac{1}{336}\sqrt{3(-112r^6+336r^5+500r^4-1560r^3-748r^2+1584r+243)}.\end{aligned}$$

Remark 3.1 Both the convolution and redistribution techniques described above are global (all particle locations and weights are changed) and are not limited to uniform auxiliary grids and allow to obtain point values of the computed particle approximation at any prescribed set of points. These methods would, in general, work perfectly fine in smooth regions, yet may smear out discontinuities that may appear in the solution or develop oscillations near nonsmooth fronts.

Remark 3.2 The time scale on which remeshing is done is mostly empirical. It is clearly correlated to the strain rate (velocity gradient) of the flow and constrained by (3.1). A simple rule that proved to be efficient in various calculations is to remesh every few time steps or even at the end of every time step. It should be observed, however, that if a redistribution procedure is applied too often, the overall resolution of the computed solution may deteriorate. Therefore, one has to come up with an (ad-hoc) strategy on when to redistribute particles, for instance, one may apply the redistribution procedure either when the smallest/largest distance between the particles becomes smaller/larger than a prescribed critical value. Additional strategies can be found in, e.g., [7, 20, 21, 23–25, 40, 41, 88, 96] and references therein.

3.2 Particle Merger — a Local Redistribution Technique

The major drawback of the aforementioned global redistribution techniques is numerical diffusion brought to the particle method every time the locations of the particles are changed and the weights are recalculated. This may be unavoidable in the case of spreading particles, but when the particles cluster, another redistribution technique — merger of clustering particles — may be used to improve the efficiency of the particle method, see, e.g., [23, 81, 109]. Particle merger techniques seem also to work perfectly well for the models with point mass concentrations and strong singularity formations (δ-functions along the surface as well as at separate points) as it was demonstrated in [23], where a sticky particle method was introduced in the context of the Euler equations of pressureless gas dynamics, [11, 112].

The simplest particle merger algorithm can be described as follows, [23]. When at a certain time t, the distance $|\boldsymbol{x}_i(t)-\boldsymbol{x}_j(t)|$ is smaller then a prescribed (ad-hoc) parameter for some i and j, then the ith and jth particles are merged into a new partcle located, say, at the center of mass of the replaced particles, i.e.,

$$\widetilde{\boldsymbol{x}}=\frac{w_i(t)\boldsymbol{x}_i(t)+w_j(t)\boldsymbol{x}_j(t)}{w_i(t)+w_j(t)}$$

and carrying the weight

$$\widetilde{w}=w_i(t)+w_j(t).$$

The cell occupied by the new particle is then set to be the union of $\Omega_i(t)$ and $\Omega_j(t)$:

$$|\widetilde{\Omega}| = |\Omega_i(t)| + |\Omega_j(t)|.$$

It should be noted that this particle merging procedure also serves as an excellent discontinuities detector — as long as the number of particles stays equal to the initial number of particles, the solution can be assumed continuous, but after the merger occurs, we assume that a discontinuity has been already formed. Moreover, by keeping a record of the location of particle, one may track the discontinuity location as well. Obviously, in large time simulations, the overall accuracy of the method may decrease due to particle mergers, since the overall number of particles decreases in time. In such a case, one would need to periodically add new particles into the smooth parts of the solution.

4 Applications to Convection-Diffusion Equations

Even though the most natural application of the particle methods is pure transport equations, over the years, the range of these methods has been extended. Many practical problems involve additional terms, besides convection (e.g. collision terms or diffusion), and hence, a demand for particle methods, which are capable to treat diffusion, dispersion and general nonlinear terms was created. As a consequence, the original particle method has been modified in a number of different ways and several approaches have been suggested for approximating derivatives as the latter became a key aspect in the development of particle methods.

The most widely used treatment of diffusion terms, the random vortex method, was introduced in [30]. In this method, the diffusion is treated by adding a Wiener process to the motion of each particle. This way the diffusion only affects the position of particles, $\boldsymbol{x}_i(t)$, while the particle weights, $w_i(t)$, remain constant in time. Numerous works followed this pioneering paper (see, e.g., [1, 5–7, 62, 64, 66, 68, 83, 87, 96]). Properties of the random vortex method have been also extensively studied in the literature. For a comprehensive list we refer to [40, 101].

Deterministic particle approximations of the derivative operators can be constructed, for instance, through the integral approximations. This can be easily achieved by taking the derivatives in equation (2.19) as convolution and derivative operators commute in unbounded or periodic domains. These approximations can be cast in a conservative formulation and have been implemented in various versions of particle methods (like, e.g., SPH [94, 95]). An alternative formulation involves the development of integral operators that are equivalent to differential operators. We review here both approaches in the context of convection-diffusion while leaving the discussion of other particle approximations out of the scope of this paper.

4.1 Particle Methods for Convection-Diffusion Equations

We consider the following convection-diffusion equation:

$$u_t + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\boldsymbol{a}u) = \nu \Delta u, \quad \nu > 0, \tag{4.1}$$

which is studied subject to the initial condition (2.2) and explore two deterministic methods for treating the diffusion terms in particle schemes.

4.1.1 Weighted Particle Method

In the so-called weighted particle method proposed in [49] (see also [42, 91]), the convective part of the equation is modeled by the convection of particles, while the diffusion part is taken into account

by changing the weights of the particles. This deterministic particle method is then based on the following integro-differential approximation of the convection-diffusion equation (4.1):

$$u_t^\sigma + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\boldsymbol{a} u^\sigma) = \nu \Delta^\sigma u^\sigma, \tag{4.2}$$

where

$$\Delta^\sigma u^\sigma := \frac{1}{\sigma^2} \int\limits_{\mathbb{R}^d} \eta_\sigma(\boldsymbol{x}-\boldsymbol{y}) \left(u(\boldsymbol{y}) - u(\boldsymbol{x})\right) d\boldsymbol{y}, \tag{4.3}$$

and the function

$$\eta_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^d} \eta\left(\frac{\boldsymbol{x}}{\sigma}\right),$$

where $\eta \in L^1(\mathbb{R}^d)$ is an even function, that is, $\eta(-\boldsymbol{x}) = \eta(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

The main idea of the weighted particle method is to apply the particle method to equation (4.2) (instead of the original equation (4.1)) and to treat the modified viscosity $\nu\Delta^\sigma u^\sigma$ as a source term. Substituting the particle expansion (2.9) into a weak formulation of (4.3), leads to the following system of ODEs for the particle locations and their weights:

$$\begin{aligned} &\frac{d\boldsymbol{x}_i(t)}{dt} = \boldsymbol{a}(\boldsymbol{x}_i(t), t), \\ &\frac{dw_i(t)}{dt} = \beta_i(t), \qquad i = 1, \ldots, N, \end{aligned}$$

where

$$\beta_i(t) = \frac{\nu}{\sigma^2} \sum_{j=1}^{N} \eta_\sigma(\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)) \left\{w_j(t)|\Omega_i(t)| - w_i(t)|\Omega_j(t)|\right\},$$

and $|\Omega_j(t)|$ is still given by (2.16).

Convergence properties of the weighted particle method have been investigated in [49], where the convergence of the solution of problem (4.2) towards the solution of (4.1) has been proven and the stability and convergence results for the weighted particle method in L^∞ have been established.

Remark 4.1 The weighted particle method has been applied to first-order systems in [51, 89, 90] and to the systems of gas dynamics in [60].

Remark 4.2 Starting from this formulation, a general deterministic integral representation for derivatives of arbitrary order was presented in [53].

Remark 4.3 A different technique in which particle methods were used for approximating solutions of the heat equation and related models (such as the Fokker-Planck equation, a Boltzmann-like equation – the Kac equation and Navier-Stokes equations), was introduced in [104, 105]. In these works, the diffusion of the particles was described as a deterministic process in terms of a mean motion with a speed equal to the osmotic velocity associated with the diffusion process. In a following work, [106], the method was shown to be successful for approximating solutions to the 2-D Navier-Stokes equation in an unbounded domain. In this setup, the particles were convected according to the velocity field while their weights evolved according to the diffusion term in the vorticity formulation of the Navier-Stokes equations.

We also refer to [59] where a different way to discretize viscous terms of the Navier-Stokes equations was suggested. The idea was to approximate the Laplacian of the vorticity by the explicit differentiation of the cut-off function. Another approach, see for example [34, 52, 53], is based on discretization of an integro-differential equation in which an integral operator approximates the diffusion operator.

4.1.2 Diffusion-Velocity Particle Method

Another deterministic approach for approximating solutions of the parabolic equations with particle methods was introduced in [50]. Their so-called diffusion-velocity method is based on defining the convective field associated with the heat operator which then allowed the particles to convect in a standard way. To this end, equation (4.1) is rewritten as a purely transport equation:

$$u_t + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\boldsymbol{a}_p u) = 0, \quad \text{where} \quad \boldsymbol{a}_p = \boldsymbol{a} - \nu \frac{\boldsymbol{\nabla}_{\boldsymbol{x}} u}{u}. \tag{4.4}$$

Here, $\boldsymbol{a}_p$ depends on u and $\nabla_{\boldsymbol{x}} u$, therefore, it can not be considered as a given function. Moreover, since the product of δ-functions is not well defined, the standard particle method has to be modified. This can be done by defining a "smoothed" velocity

$$\boldsymbol{a}_\zeta(u) = \boldsymbol{a} - \nu \frac{u * \boldsymbol{\nabla}_{\boldsymbol{x}} \zeta_\varepsilon}{u * \zeta_\varepsilon}, \tag{4.5}$$

where ζ_ε is a cut-off function defined in (2.18). Equation (4.4) is then replaced by the following transport equation

$$u_t + \boldsymbol{\nabla}_{\boldsymbol{x}} (\boldsymbol{a}_\zeta(u) u) = 0, \tag{4.6}$$

which is called the diffusion-velocity transport equation. The resulting diffusion-velocity particle method is obtained by considering a particle approximation as a distribution of the form (2.19), where $x_i(t)$ and $w_i(t)$ are the solutions to

$$\begin{cases} \dfrac{d\boldsymbol{x}_i}{dt} = \boldsymbol{a}_\zeta(u_\varepsilon^N(\boldsymbol{x}_i(t), t)) = \boldsymbol{a}(\boldsymbol{x}_i(t), t) - \nu \dfrac{\sum\limits_{j=1}^N w_j \boldsymbol{\nabla}_{\boldsymbol{x}} \zeta_\varepsilon(\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t))}{\sum\limits_{j=1}^N w_j \zeta_\varepsilon(\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t))}, \\ \dfrac{dw_i(t)}{dt} = 0, \end{cases} \qquad i = 1, \dots, N. \tag{4.7}$$

Notice that in this method, the computed solution u_ε^N should be recovered from its particle distributions at every time step during the time integration (4.7) (unlike for linear transport equations, where the solution is recovered only at the final time).

The convergence properties of the diffusion-velocity particle method were investigated, e.g., in [78, 79], where short time existence and uniqueness of solutions for the resulting diffusion-velocity transport equation were proved. Convergence results for a porous-media equation were also obtained in [86].

Acknowledgment. This work was supported in part by the NSF Grant DMS-1521051. The author thanks A. Kurganov for the constructive discussions.

References

[1] C. Anderson and C. Greengard, *On vortex methods*, SIAM J. Numer. Anal. **22** (1985), no. 3, 413–440.

[2] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput. **13** (1992), no. 4, 923–947.

[3] V. I. Arnold, *Ordinary differential equations*, Universitext, Springer-Verlag, Berlin, 2006, Translated from the Russian by Roger Cooke, Second printing of the 1992 edition.

[4] J. Barnes and P. Hut, *A hierarchical O(N log N) force-calculation algorithm*, nature **324** (1986), no. 6096, 446–449.

[5] J. T. Beale and A. Majda, *Vortex methods. I. Convergence in three dimensions*, Math. Comp. **39** (1982), no. 159, 1–27.

[6] ———, *Vortex methods. II. Higher order accuracy in two and three dimensions*, Math. Comp. **39** (1982), no. 159, 29–52.

[7] ———, *High order accurate vortex methods with explicit velocity kernels*, J. Comput. Phys. **58** (1985), 188–208.

[8] M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos, *Multilevel adaptive particle methods for convection-diffusion equations*, Multiscale Model. Simul. **4** (2005), no. 1, 328–357.

[9] M. Bergdorf and P. Koumoutsakos, *A Lagrangian particle-wavelet method*, Multiscale Model. Simul. **5** (2006), no. 3, 980–995.

[10] ———, *Multiresolution particle methods*, Complex effects in large eddy simulations, Lect. Notes Comput. Sci. Eng., vol. 56, Springer, Berlin, 2007, pp. 49–61.

[11] Y. Brenier and E. Grenier, *Sticky particles and scalar conservation laws*, SIAM J. Numer. Anal. **35** (1998), 2317–2328.

[12] R. Camassa, J. Huang, and L. Lee, *On a completely integrable numerical scheme for a nonlinear shallow-water wave equation*, J. Nonlinear Math. Phys. **12** (2005), no. suppl. 1, 146–162.

[13] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput. **9** (1988), no. 4, 669–686.

[14] P. Chatelain, G.-H. Cottet, and P. Koumoutsakos, *Particle mesh hydrodynamics for astrophysics simulations*, Internat. J. Modern Phys. C **18** (2007), no. 4, 610–618.

[15] A. Chertock, C. R. Doering, E. Kashdan, and A. Kurganov, *A fast explicit operator splitting method for passive scalar advection*, J. Sci. Comput. **45** (2010), 200–214.

[16] A. Chertock, P. Du Toit, and J. E. Marsden, *Integration of the EPDiff equation by particle methods*, ESAIM Math. Model. Numer. Anal. **46** (2012), no. 3, 515–534.

[17] A. Chertock, E. Kashdan, and A. Kurganov, *Propagation of diffusing pollutant by a hybrid Eulerian-Lagrangian method*, Hyperbolic problems: theory, numerics, applications, Springer, Berlin, 2008, pp. 371–379.

[18] A. Chertock and A. Kurganov, *On a hybrid finite-volume particle method*, M2AN Math. Model. Numer. Anal **38** (2004), 1071–1091.

[19] ———, *Conservative locally moving mesh method for multifluid flows*, Finite Volumes for Complex Applications IV (F. Benkhaldoun, D. Ouazar, and S. Raghay, eds.), Hermes Science, 2005, pp. 273–284.

[20] ———, *On a practical implementation of particle methods*, Appl. Numer. Math. **56** (2006), 1418–1431.

[21] ———, *Computing multivalued solutions of pressureless gas dynamics by deterministic particle methods*, Commun. Comput. Phys. **5** (2009), 565–581.

[22] A. Chertock, A. Kurganov, and G. Petrova, *Finite-volume-particle methods for models of transport of pollutant in shallow water*, J. Sci. Comput. **27** (2006), 189–199.

[23] A. Chertock, A. Kurganov, and Yu. Rykov, *A new sticky particle method for pressureless gas dynamics*, SIAM J. Numer. Anal. **45** (2007), 2408–2441.

[24] A. Chertock and D. Levy, *Particle methods for dispersive equations*, J. Comput. Phys. **171** (2001), no. 2, 708–730.

[25] ———, *A particle method for the KdV equation*, J. Sci. Comput. **17** (2002), 491–499.

[26] A. Chertock, J.-G. Liu, and T. Pendleton, *Convergence analysis of the particle method for the Camassa-Holm equation*, Proceedings of the 13th International Conference on "Hyperbolic Problems: Theory, Numerics and Applications" (Ph. G. Ciarlet and Ta-Tsien Li, eds.), Contemporary Applied Mathematics, 2012, pp. 356–373.

[27] ———, *Convergence of a particle method and global weak solutions of a family of evolutionary PDEs*, SIAM J. Numer. Anal. **50** (2012), no. 1, 1–21.

[28] ———, *Elastic collisions among peakon solutions for the Camassa-Holm equation*, Appl. Numer. Math. **93** (2015), 30–46.

[29] J. P. Choquin and S. Huberson, *Particle simulation of viscous flow*, Comput. & Fluids **17** (1989), no. 2, 397–410.

[30] A. J. Chorin, *Numerical study of slightly viscous flow*, J. Fluid Mech. **57** (1973), no. 4, 785–796.

[31] A. J. Chorin and J. E. Marsden, *A mathematical introduction to fluid mechanics*, third ed., Texts in Applied Mathematics, vol. 4, Springer-Verlag, New York, 1993.

[32] A. Cohen and B. Perthame, *Optimal approximations of transport equations by particle and pseudoparticle methods*, SIAM J. Math. Anal. **32** (2000), no. 3, 616–636.

[33] M. H. Cohen and A. Robertson, *Wave propagation in the early stages of aggregation of cellular slime molds*, J. Theor. Biol. **31** (1971), 101–118.

[34] R. Cortez, *Convergence of high-order deterministic particle methods for the convection-diffusion equation*, Comm. Pure Appl. Math. **50** (1997), no. 12, 1235–1260.

[35] G.-H. Cottet, *A particle-grid superposition method for the Navier-Stokes equations*, J. Comput. Phys. **89** (1990), no. 2, 301–318.

[36] ———, *A particle model for fluid-structure interaction*, C. R. Math. Acad. Sci. Paris **335** (2002), no. 10, 833–838.

[37] G.-H. Cottet, G. Balarac, and M. Coquerelle, *Subgrid particle resolution for the turbulent transport of a passive scalar*, pp. 779–782, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[38] G.-H. Cottet, J.-M. Etancelin, F. Perignon, and C. Picard, *High order semi-Lagrangian particle methods for transport equations: numerical analysis and implementation issues*, ESAIM Math. Model. Numer. Anal. **48** (2014), no. 4, 1029–1060.

[39] G.-H. Cottet, P. Koumoutsakos, and M. L. O. Salihi, *Vortex methods with spatially varying cores*, J. Comput. Phys. **162** (2000), no. 1, 164–185.

[40] G.-H. Cottet and P. D. Koumoutsakos, *Vortex methods*, Cambridge University Press, Cambridge, 2000.

[41] G.-H. Cottet and A. Magni, *TVD remeshing formulas for particle methods*, C. R. Math. Acad. Sci. Paris **347** (2009), no. 23-24, 1367–1372.

[42] G.-H. Cottet and S. Mas-Gallic, *A particle method to solve transport-diffusion equations, part 1: the linear case*, Tech. Report 115, Ecole Polytechnique, Palaiseau, France, 1983.

[43] ———, *A particle method to solve the Navier-Stokes system*, Numer. Math. **57** (1990), no. 8, 805–827.

[44] G.-H. Cottet and P. Poncet, *Advances in direct numerical simulations of 3D wall-bounded flows by vortex-in-cell methods*, J. Comput. Phys. **193** (2004), no. 1, 136–158.

[45] G.-H. Cottet and L. Weynans, *Particle methods revisited: a class of high order finite-difference methods*, C. R. Math. Acad. Sci. Paris **343** (2006), no. 1, 51–56.

[46] S. Cui, A. Kurganov, and A. Medovikov, *Particle methods for PDEs arising in financial modeling*, Appl. Numer. Math. (2015), 123–139.

[47] A. Degasperis, D. D. Holm, and A. N. W. Hone, *A new integrable equation with peakon solutions*, Theoret. Math. Phys. **133** (2002), 1463–1474.

[48] A. Degasperis and M. Procesi, *Asymptotic integrability*, Symmetry and perturbation theory (Rome, 1998), World Sci. Publ., River Edge, NJ, 1999, pp. 23–37.

[49] P. Degond and S. Mas-Gallic, *The weighted particle method for convection-diffusion equations. part 1 and part 2*, Math. Comp. **53** (1989), 485–525.

[50] P. Degond and F. J. Mustieles, *A deterministic approximation of diffusion equations using particles*, SIAM J. Sci. Stat. Comp. **11** (1990), no. 2, 293–310.

[51] P. Degond and B. Niclot, *Numerical analysis of the weighted particle method applied to the semiconductor boltzmann equation*, Numer. Math. **55** (1989), no. 5, 599–618.

[52] J. D. Eldredge, T. Colonius, and A. Leonard, *A vortex particle method for two-dimensional compressible flow*, J. Comput. Phys. **179** (2002), no. 2, 371–399.

[53] J. D. Eldridge, A. Leonard, and T. Colonious, *A general deterministic treatment of derivatives in particle methods*, J. Comput. Phys. **180** (2002), no. 2, 686–709.

[54] B. Engquist and T. Y. Hou, *Particle method approximation of oscillatory solutions to hyperbolic differential equations*, SIAM J. Numer. Anal. **26** (1989), no. 2, 289–319.

[55] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys. **183** (2002), no. 1, 83–116.

[56] D. Enright, F. Losasso, and R. Fedkiw, *A fast and accurate semi-Lagrangian particle level set method*, Comput. & Structures **83** (2005), no. 6-7, 479–490.

[57] L. C. Evans, *Partial differential equations*, Graduate Studies in Mathematics, vol. 19, American Mathematical Society, Providence, RI, 1998.

[58] M. W. Evans and F. H. Harlow, *The particle-in-cell method for hydrodynamic calculations*, Tech. Report LA-2139, Los Alamos Scientific Laboratory, 1957.

[59] D. Fishelov, *A new vortex scheme for viscous flows*, J. Comput. Phys. **86** (1990), 211–224.

[60] R. J. Gingold and J. J. Monaghan, *Shock simulation by the particle method S.P.H.*, J. Comput. Phys. **37** (1987), 374–389.

[61] R. T. Glassey, *The Cauchy problem in kinetic theory*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[62] J. Goodman, *Convergence of the random vortex method*, Comm. Pure Appl. Math. **40** (1987), no. 2, 189–220.

[63] L. Greengard and V. Rokhlin, *The rapid evaluation of potential fields in three dimensions*, Vortex methods (Los Angeles, CA, 1987), Lecture Notes in Math., vol. 1360, Springer, Berlin, 1988, pp. 121–141.

[64] K. Gustafson and J. Sethian, *Vortex methods and vortex motion*, SIAM, Philadelphia, 1991.

[65] O. Hald and V. M. del Prete, *Convergence of vortex methods for Euler's equations*, Math. Comp. **32** (1978), no. 143, 791–809.

[66] O. H. Hald, *Convergence of vortex methods for Euler's equations. II*, SIAM J. Numer. Anal. **16** (1979), no. 5, 726–755.

[67] ———, *Convergence of vortex methods for Euler's equations. III*, SIAM J. Numer. Anal. **24** (1987), no. 3, 538–582.

[68] ———, *Convergence of vortex methods*, Vortex methods and vortex motion, SIAM, Philadelphia, PA, 1991, pp. 33–58.

[69] F. H. Harlow, *A machine calculation method for hydrodynamic problems*, Tech. Report LAMS-1956, Los Alamos Scientific Laboratory, 1956.

[70] ———, *The particle-in-cell computing method for fluid dynamics*, Methods in Computational Physics (B. Alder, S. Fernbach, and M. Rotenberg, eds.), vol. 3, Academic Press, New York, 1964, pp. 319–343.

[71] F. Hermeline, *A deterministic particle method for transport diffusion equations: Application to the fokker-planck equation*, J. Comput. Phys **82** (1989), no. 1, 122–146.

[72] S. E. Hieber and P. Koumoutsakos, *A Lagrangian particle level set method*, J. Comput. Phys. **210** (2005), no. 1, 342–367.

[73] R. W Hockney and J. W Eastwood, *Computer simulation using particles*, CRC Press, 2010.

[74] T. Y. Hou, *Convergence of a variable blob vortex method for the Euler and Navier-Stokes equations*, SIAM J. Numer. Anal. **27** (1990), no. 6, 1387–1404.

[75] P. Koumoutsakos, *Multiscale flow simulations using particles*, Annual review of fluid mechanics. Vol. 37, Annu. Rev. Fluid Mech., vol. 37, Annual Reviews, Palo Alto, CA, 2005, pp. 457–487.

[76] P. Koumoutsakos and A. Leonard, *High-resolution simulations of the flow around an impulsively started cylinder using vortex methods*, Journal of Fluid Mechanics **296** (1995), 1–38.

[77] R. Krasny, *A study of singularity formation in a vortex sheet by the point-vortex approximation*, J. Fluid Mech. **167** (1986), 65–93.

[78] G. Lacombe, *Analyse d'une équation de vitesse de diffusion*, C. R. Acad. Sci. Paris Sér. I Math. **329** (1999), no. 5, 383–386.

[79] G. Lacombe and S. Mas-Gallic, *Presentation and analysis of a diffusion-velocity method*, Flows and related numerical methods (Toulouse, 1998), ESAIM Proc., vol. 7, Soc. Math. Appl. Indust., Paris, 1999, pp. 225–233 (electronic).

[80] J.-B. Lagaert, G. Balarac, and G.-H. Cottet, *Hybrid spectral-particle method for the turbulent transport of a passive scalar*, J. Comput. Phys. **260** (2014), 127–142.

[81] G. Lapenta, *Particle rezoning for multidimensional kinetic particle-in-cell simulations*, J. Comput. Phys. **181** (2002), 317–337.

[82] A. Leonard, *Vortex methods for flow simulation*, J. Comput. Phys. **37** (1980), no. 3, 289–335.

[83] ———, *Computing three-dimensional incompressible flows with vortex elements*, Ann. Rev. Fluid Mech. **17** (1985), 289–335.

[84] S. Li and W. K. Liu, *Meshfree particle methods*, Springer-Verlag, Berlin, 2004.

[85] K. Lindsay and R. Krasny, *A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow*, J. Comput. Phys. **172** (2001), no. 2, 879–907.

[86] P.-L. Lions and S. Mas-Gallic, *Une méthode particulaire déterministe pour des équations diffusives non linéaires*, C. R. Acad. Sci. Paris Sér. I Math. **332** (2001), no. 4, 369–376.

[87] D.-G. Long, *Convergence of the random vortex method in two dimensions*, J. Amer. Math. Soc. **1** (1988), no. 4, 779–804.

[88] A. Magni and G.-H. Cottet, *Accurate, non-oscillatory, remeshing schemes for particle methods*, J. Comput. Phys. **231** (2012), no. 1, 152–172.

[89] S. Mas-Gallic, *A deterministic particle method for the linearized boltzmann equation*, Transport Theory and Stat. Phys. **16** (1987), 855–887.

[90] S. Mas-Gallic and F. Poupaud, *Approximation of the transport equation by a weighted particle method*, Transport Theory and Stat. Phys. **17** (1988), 311–345.

[91] S. Mas-Gallic and P. A. Raviart, *Particle approximation of convection-diffusion equations*, International Report 86013, Université Pierre et Marie Curie, 1986.

[92] R. McLachlan and S. Marsland, *N-particle dynamics of the Euler equations for planar diffeomorfism*, Dyn. Syst. **22** (2007), no. 3, 269–290.

[93] J. J. Monaghan, *Extrapolating B-splines for interpolation*, J. Comput. Phys. **60** (1985), no. 2, 253–262.

[94] ———, *Particle methods for hydrodynamics*, Comp. Phys. Rep. **3** (1985), no. 2, 71–124.

[95] ———, *An introduction to sph*, Comp. Phys. Commun. **48** (1988), no. 1, 89–96.

[96] H. Nordmark, *Rezoning for higher order vortex methods*, J. Comput. Phys. **97** (1991), 366–397.

[97] M. L. Ould-Salihi, G.-H. Cottet, and M. El Hamraoui, *Blending finite-difference and vortex methods for incompressible flow computations*, SIAM J. Sci. Comput. **22** (2000), no. 5, 1655–1674 (electronic).

[98] M. Perlman, *On the accuracy of vortex methods*, J. Comput. Phys. **59** (1985), no. 2, 200–223.

[99] C. S. Peskin, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517.

[100] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren, *Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at* $\mathrm{Re} = 300,\ 500$*, and 1000*, J. Comput. Phys. **178** (2002), no. 2, 427–463.

[101] E. G. Puckett, *Vortex methods - an introduction survey of selected research topics*, Incompressible computational fluid dynamics - Trends and advances (M.D. Gunzburger and R.A. Nicolaides, eds.), Cambridge University Press, 1993, pp. 335–407.

[102] J. T. Rasmussen, G.-H. Cottet, and J. H. Walther, *A multiresolution remeshed Vortex-In-Cell algorithm using patches*, J. Comput. Phys. **230** (2011), no. 17, 6742–6755.

[103] P.-A. Raviart, *An analysis of particle methods*, Numerical methods in fluid dynamics (Como, 1983), Lecture Notes in Math., vol. 1127, Springer, Berlin, 1985, pp. 243–324.

[104] G. Russo, *Deterministic diffusion of particles*, Comm. Pure Appl. Math. **43** (1990), 697–733.

[105] ———, *A particle method for collisional kinetic equations. I. Basic theory and one-dimensional results*, J. Comput. Phys. **87** (1990), 270–300.

[106] ———, *A deterministic vortex method for the Navier-Stokes equations*, J. Comput. Phys. **108** (1993), 87–95.

[107] I. J. Schoenberg, *Cardinal spline interpolation*, Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1973, Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics, No. 12.

[108] J. Strain, $2D$ *vortex methods and singular quadrature rules*, J. Comput. Phys. **124** (1996), no. 1, 131–145.

[109] J. Teunissen and U. Ebert, *Controlling the weights of simulation particles: adaptive particle management using* k*-d trees*, J. Comput. Phys. **259** (2014), 318–330.

[110] W. M. van Rees, A. Leonard, D. I. Pullin, and P. Koumoutsakos, *A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers*, J. Comput. Phys. **230** (2011), no. 8, 2794–2805.

[111] D. Wee and A. F. Ghoniem, *Modified interpolation kernels for treating diffusion and remeshing in vortex methods*, J. Comput. Phys. **213** (2006), no. 1, 239–263.

[112] Ya. B. Zeldovich, *Gravitational instability: An approximate theory for large density perturbations*, Astron. Astrophys. **5** (1970), 84–89.