



**HAL**  
open science

# A Review of Vortex Methods and Their Applications: From Creation to Recent Advances

Chloé Mimeau, Iraj Mortazavi

► **To cite this version:**

Chloé Mimeau, Iraj Mortazavi. A Review of Vortex Methods and Their Applications: From Creation to Recent Advances. *Fluids*, 2021, 6 (2), pp.68. 10.3390/fluids6020068 . hal-03677567

**HAL Id: hal-03677567**

**<https://cnam.hal.science/hal-03677567>**

Submitted on 24 May 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Review

# A Review of Vortex Methods and Their Applications: From Creation to Recent Advances

Chloé Mimeau \*  and Iraj Mortazavi 

Laboratory M2N, CNAM, 75003 Paris, France; iraj.mortazavi@cnam.fr

\* Correspondence: chloe.mimeau@cnam.fr; Tel.: +33-0140-272-283

**Abstract:** This review paper presents an overview of Vortex Methods for flow simulation and their different sub-approaches, from their creation to the present. Particle methods distinguish themselves by their intuitive and natural description of the fluid flow as well as their low numerical dissipation and their stability. Vortex methods belong to Lagrangian approaches and allow us to solve the incompressible Navier-Stokes equations in their velocity-vorticity formulation. In the last three decades, the wide range of research works performed on these methods allowed us to highlight their robustness and accuracy while providing efficient computational algorithms and a solid mathematical framework. On the other hand, many efforts have been devoted to overcoming their main intrinsic difficulties, mostly relying on the treatment of the boundary conditions and the distortion of particle distribution. The present review aims to describe the Vortex methods by following their chronological evolution and provides for each step of their development the mathematical framework, the strengths and limits as well as references to applications and numerical simulations. The paper ends with a presentation of some challenging and very recent works based on Vortex methods and successfully applied to problems such as hydrodynamics, turbulent wake dynamics, sediment or porous flows.

**Keywords:** hydrodynamics; incompressible flows; particle methods; remeshing; semi-Lagrangian vortex methods; vortex methods; Vortex-Particle-Mesh methods; vorticity; wake dynamics



**Citation:** Mimeau, C.; Mortazavi, I. A Review of Vortex Methods and Their Applications: From Creation to Recent Advances. *Fluids* **2021**, *6*, 68. <https://doi.org/10.3390/fluids6020068>

Academic Editor: Markus Klein  
Received: 24 December 2020  
Accepted: 28 January 2021  
Published: 4 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Vortex methods (VM) belong to Lagrangian methods, also called particle methods, used to solve continuous systems of the form:

$$\frac{d}{dt} \int_{\Omega} \mathbf{Q}(\mathbf{x}, t) d\mathbf{x} = \int_{\Omega} \mathbf{F}(\mathbf{x}, t, \mathbf{Q}, \nabla \mathbf{Q}, \dots) d\mathbf{x} \quad (1)$$

where  $\mathbf{Q}$  denotes the unsteady solution vector evolving in space ( $\mathbf{x}$ ) and time ( $t$ ), composed of the physical quantities  $Q_i$ , and where  $\mathbf{F}$  is the source term depending on space and time and on the solution  $\mathbf{Q}$  and its partial derivatives. In a general point of view, Lagrangian methods differ from Eulerian ones by the way they discretize the problem. In an Eulerian approach, the computational domain is described as a mesh and the solution vector  $\mathbf{Q}$  is approximated on each node according to a numerical scheme involving the solution at the neighboring mesh nodes. In the Lagrangian framework, the domain discretization is not realized on a mesh but on a set of particles that follow the dynamic of the system and are displaced with respect to the flow velocity  $\mathbf{u}$ . More precisely, in the Lagrangian approach, the solution  $\mathbf{Q}$  is discretized on numerical particles and, given a location  $\mathbf{x}$  in the domain, each quantity  $Q$  among the vector solution  $\mathbf{Q}$  is thus approximated at this location  $\mathbf{x}$  by:

$$Q(\mathbf{x}) = \int_{\Omega} Q(\mathbf{y}) \delta(\mathbf{x} - \mathbf{y}) d\mathbf{y} \quad (2)$$

where  $\delta$  denotes the Dirac distribution. To write this expression in a regularized and discrete form, the Lagrangian methods define a volume  $v_p$  for the particles as well as a

smoothing function  $W$ , of compact support and radial symmetry of radius  $\varepsilon$ , which tends to the Dirac distribution when  $\varepsilon$  tends to 0. Therefore, the discrete approximation of the quantity  $Q$  at location  $\mathbf{x}$  can be written as:

$$Q(\mathbf{x}) \simeq \sum_p Q_p W_\varepsilon(\mathbf{x} - \mathbf{x}_p) \quad (3)$$

In other words, the quantity  $Q$  at location  $\mathbf{x}$  is evaluated as the sum of the quantity  $Q$  carried by the particles  $p$  located in the compact support of the smoothing radial function  $W$  of radius  $\varepsilon$ , centered at location  $\mathbf{x}$ . The Lagrangian discretization of the continuous system (1) is therefore obtained by solving the following system of ordinary differential equations (ODEs) in order to update the particle positions  $\mathbf{x}_p$ , their volume  $v_p$  and each quantity  $Q_p$  they carry according to the source term  $\mathbf{F}$ :

$$\begin{cases} \frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p, t) \\ \frac{dv_p}{dt} = v_p \operatorname{div}(\mathbf{u}(\mathbf{x}_p)) \\ \frac{dQ_p}{dt} = v_p \mathbf{F}_p \end{cases} \quad (4)$$

The key common thread of all particle methods used for continuum flow simulations therefore relies on the approximation in the Lagrangian form of the governing continuous Equation (1) where the derivative operators involved in  $\mathbf{F}$  are replaced by equivalent integral operators that are then discretized on the particle locations. Among the most popular particle methods used to solve continuous equations, one can cite the Vortex Methods (VM) and the Smooth Particle Hydrodynamics (SPH). They differ from each other by the governing Equation (1) they aim to solve and by the way they handle the resolution of the third ODE of system (4). SPH methods [1–5] solve the weakly compressible Navier-Stokes equations in velocity-pressure formulation and the particles carry the pressure and the flow density variables. They need a coupling with an equation of state to close the governing system. Regarding Vortex Methods, they are used to solve the incompressible Navier-Stokes equations in their velocity-vorticity formulation and the particles only carry the vorticity field.

If one starts from a present observation, one admits that in the computational fluid dynamic community, Vortex Methods do not benefit from the same recognition or the same expanded scope than Eulerian methods (spectral methods, finite elements, finite volumes, finite differences, etc.). However, they early raised the interest of the CFD research community due to their capability to precisely and naturally simulate flows dominated by the advection phenomenon. They have been the subject of plentiful of research works, which allowed these last recent years to lead to challenging applications involving important issues in fluid dynamics, in particular in hydrodynamics [6–9], in flow mediated transport [10,11], in turbulent wake dynamics of rotorcrafts [12,13], aircrafts [13,14] and vertical axis wind turbines [15,16], in aeroacoustics [17], in free jet flows [18] or in active [19] and passive [20] flow control. One of the first thorough reviews of Vortex methods has been proposed in 1991 by Sethian and Gustafson [21]. In 2000, Cottet and Koumoutsakos [22] proposed a further detailed book, gathering all the theory and numerical practice related to these methods. In 2011, Yokota and Obi completed this overview with a review paper dedicated to the simulation of turbulent flows with Vortex methods [23].

The present paper, starting from preliminary survey carried out in [24], proposes a synthetic review of Vortex methods from their genesis in the 1930s until their very recent advances. In the context of incompressible flow simulations, one of the first goals of this work is to provide an insight of a particle approach based on a solid mathematical framework with convergence analysis and conservation features, which is able to provide accurate simulations and which could be considered as a complement or an alternative to purely Eulerian methods. Concerning the Vortex methods themselves, the literature

survey of this review aims to give an overall landscape (although not exhaustive due to the finite format of a review paper) of the existing Vortex Methods, depicting the different sub-families of VMs with their own advantages, drawbacks and applications. It also aims to highlight the intrinsic advantages of Vortex Methods that explain their intensive and continuous development from the 1970s, and attempts at showing how they progressively evolved in time to meet the requirements involved by the CFD community, namely the capability of handling complex physics while ensuring convergence to solution, accuracy, efficiency, and performance. This review is addressed both to scientists specialized in numerical analysis and discretization schemes, and those dedicated to incompressible fluid dynamics using as well direct numerical simulations or large eddy simulations.

The present review paper follows the chronological advances of Vortex Methods and is structured this way: First, Section 2 gives the global description and formulation of Vortex Methods in the initial context they were used for, that is to say the resolution of the inviscid Euler equations. In Section 3, one will explain how they expand to solve the Navier-Stokes equations, including the diffusion effects. Section 4 will address the crucial issue of particle distortion in Lagrangian methods and their need to perform a spatial adaptation of the particle locations in order to ensure the numerical convergence; this section introduces the so-called hybrid Vortex Methods, which include in their discretization schemes the presence of an underlying grid in order to circumvent the particle distortion phenomenon. In Section 5, we will focus on the treatment of boundary conditions in VMs, and more precisely on the enforcement of no-slip boundary conditions at the surface of a solid and rigid immersed obstacle. Section 6 will describe the progresses made in VMs in terms of algorithmic issues, allowing them to handle complex or computationally costly problems. Finally, Section 7 will propose a finite list of relevant and challenging simulations performed with Vortex Methods.

## 2. Vortex Methods for Inviscid Incompressible Flows

### 2.1. General Description and Formulation

The first calculations based on vortex methods were performed, by hand, independently by Prager in 1928 [25] and Rosenhead in 1931 [26]. For that purpose, they discretized the flow into elements, namely the particles, characterized by their position and the local circulation they carry. One recall that the circulation  $\Gamma$  corresponds to the macroscopic rotation of the flow. It is a scalar field defined as the integral of the vorticity field  $\omega$  on patch of vorticity  $\mathcal{A}$ , bounded by a closed path:

$$\Gamma = \int_{\mathcal{A}} \omega \, d\mathbf{x}, \quad (5)$$

where the vorticity  $\omega$  measures the local rotation of the fluid flow and is obtained by taking the curl of the velocity field:

$$\omega \equiv \nabla \times \mathbf{u}. \quad (6)$$

In the case of a two dimensional flow, the vorticity is perpendicular to the flow plane and corresponds to a scalar field. We note it by  $\omega$  in that case. In the Cartesian coordinates system, the vorticity respectively expresses in 2D and 3D as follows:

$$\omega = \omega_z = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}, \quad \omega = (\omega_w, \omega_y, \omega_z) = \left( \frac{\partial u_z}{\partial y} - \frac{\partial u_y}{\partial z}, \frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x}, \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right). \quad (7)$$

Based on the pioneer works [25,26], the method described in the sequel corresponds to the general definition of vortex methods, first introduced in the 1970s for the two-dimensional formulation [27] and in the late 1980s for the three-dimensional formulation [28,29].

The design of vortex methods for inviscid incompressible flows is based on vorticity-velocity formulation of the Euler equations. The latter represent hyperbolic equations of conservation of mass (continuity), and balance of momentum and energy. They correspond

to particular Navier–Stokes equations for inviscid (zero viscosity) flows, without thermal conductivity. In vorticity-velocity formulation and in the case of incompressible flows, they can be written as follows:

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla) \omega - (\omega \cdot \nabla) \mathbf{u} = 0 \tag{8}$$

$$\text{div } \mathbf{u} = 0 \tag{9}$$

$$\omega(\cdot, 0) = \omega_0 \tag{10}$$

$$|\mathbf{u}| \rightarrow \mathbf{u}_\infty \tag{11}$$

where  $(\mathbf{u} \cdot \nabla) \omega$  denotes the advection term,  $(\omega \cdot \nabla) \mathbf{u} = [\nabla \mathbf{u}][\omega]$  corresponds to the stretching term (which vanishes in two-dimensions) and where Equation (9) corresponds to the incompressibility condition, resulting from the continuity equation. In fact, the Euler equations were originally formulated in the Lagrangian form which involves  $D\omega/dt$ , the material derivative in time of the vorticity  $\omega$  with respect to the advective field  $\mathbf{u}$ . The discretization of these equations with a vortex method, that is to say with a particle method, is therefore natural and straightforward. It goes as follows: Let a regular approximation  $\omega^h$  of  $\omega$ , and let  $\mathbf{x}_p^h$  the trajectories of the particles transported by the flow, the approximated vorticity  $\omega^h$  may be numerically expressed at time  $t$  by:

$$\omega^h(\mathbf{x}, t) = \sum_p \alpha_p^h \delta(\mathbf{x} - \mathbf{x}_p^h(t)), \tag{12}$$

where the numerical particles locations  $\mathbf{x}_p^h(t)$  are obtained by solving the following ODEs:

$$\frac{d\mathbf{x}_p^h(t)}{dt} = \mathbf{u}^h(\mathbf{x}_p^h(t), t) \tag{13}$$

and where  $\alpha_p^h$  corresponds to the local circulation around  $\mathbf{x}_p^h$ , defined as the product of the vorticity by the volume of the particle  $\alpha_p^h = \mathbf{v}_p \omega_p^h$ . According to the Kelvin’s theorem, which states that in an inviscid flow the circulation around a closed curve remains constant in time when moving with the fluid, the local circulation  $\alpha_p^h$  satisfies the following ODEs in 2D and 3D respectively:

$$\frac{d\alpha_p^h}{dt} = 0 \text{ (2D)}, \quad \frac{d\alpha_p^h}{dt} = \alpha_p^h \cdot \nabla \mathbf{u}^h(\mathbf{x}_p^h, t) \text{ (3D)}. \tag{14}$$

From the Kelvin’s and Helmholtz theorems, the vorticity is advected at the local fluid velocity  $\mathbf{u}^h$  in the potential flow (one refers the reader to section 1.3 of Cottet and Koumoutsakos’ book [22] for a detailed presentation of the first and second Helmholtz theorems that drive the motions of fluid particles carrying vorticity in inviscid flows and their further extension by Kelvin to include the effects of viscosity). The velocity can be recovered at all times from the vorticity through the following Poisson equation, based on the incompressibility condition:

$$\Delta \mathbf{u}^h = -\nabla \times \omega^h, \tag{15}$$

which can be solved through an integral representation:

$$\mathbf{u}^h(\mathbf{x}) = \text{curl}(\mathbf{G}(\mathbf{x}, \mathbf{x}') \star \omega^h(\mathbf{x})) + \mathbf{u}_\infty^h(\mathbf{x}), \tag{16}$$

where  $\mathbf{G}$  is the Green’s function for the Laplacian operator,  $\nabla^2 \mathbf{G}(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$  and where  $\star$  denotes the convolution product. Then, from Equation (16) and denoting  $\mathbf{K} = \nabla \mathbf{G}$ , one finally obtains the velocity field through the following Biot–Savart equation:

$$\mathbf{u}^h = \mathbf{K} \star \omega^h + \mathbf{u}_\infty^h, \tag{17}$$

with

$$\text{in 2D : } \quad \mathbf{G}(\mathbf{x}) = \frac{1}{2\pi} \ln(|\mathbf{x}|), \quad \mathbf{K}(\mathbf{x}, \mathbf{x}') = \nabla \mathbf{G}(\mathbf{x}, \mathbf{x}') = \frac{1}{2\pi} \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^2} \quad (18)$$

$$\text{in 3D : } \quad \mathbf{G}(\mathbf{x}) = (4\pi|\mathbf{x}|)^{-1}, \quad \mathbf{K}(\mathbf{x}, \mathbf{x}') = \nabla \mathbf{G}(\mathbf{x}, \mathbf{x}') = -\frac{1}{4\pi} \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} \quad (19)$$

To summarize, in vortex method, each vortex element is made to move at a single velocity according to the Biot–Savart law and Equations (12)–(14) and (17) model the transport of the vorticity in inviscid incompressible flow. However, the Dirac distribution involved in the vorticity definition (12) makes unaffordable the numerical discretization and approximation of the vorticity field in practice. Faced with this problem arose the idea of the vortex blob method, originally introduced by Chorin and Bernard [27]. In this numerical approach, the vorticity of each particle is distributed on a blob, that is to say on a disk of finite radius  $\varepsilon$ . This vorticity blob-distribution is defined by a smooth cutoff function  $\zeta$ .

If the blob radius  $\varepsilon$  is small enough, one denotes  $\zeta_\varepsilon$  the mollified function defined in 2D and 3D by:

$$\zeta_\varepsilon = \frac{1}{\varepsilon^n} \zeta\left(\frac{\mathbf{x}}{\varepsilon}\right), \quad n = 2, 3. \quad (20)$$

Among the possible cutoff functions, the Gaussian-based ones are often favored because of their smoothness property, their radial symmetry and their fast decay. In vortex blob method, the vorticity  $\omega_\varepsilon^h$  of the mollified particles is then given by the following expression (one refers to Figure 1 for a graphical support):

$$\omega_\varepsilon^h(\mathbf{x}) = \sum_p \alpha_{p_\varepsilon}^h \zeta_\varepsilon(\mathbf{x} - \mathbf{x}_p^h), \quad (21)$$

with

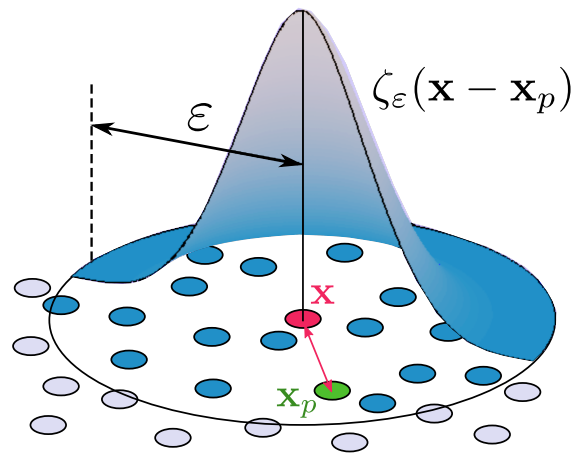
$$\frac{d\alpha_{p_\varepsilon}^h}{dt} = 0 \quad (2D), \quad / \quad \frac{d\alpha_{p_\varepsilon}^h}{dt} = \alpha_{p_\varepsilon}^h \cdot \nabla \mathbf{u}_\varepsilon^h(\mathbf{x}_p^h, t) \quad (3D) \quad (22)$$

$$\frac{d\mathbf{x}_p^h(t)}{dt} = \mathbf{u}_\varepsilon^h(\mathbf{x}_p^h(t), t) \quad (23)$$

$$\mathbf{u}_\varepsilon^h = \mathbf{K}_\varepsilon \star \omega_\varepsilon^h = (\mathbf{K} \star \zeta_\varepsilon) \star \omega_\varepsilon^h. \quad (24)$$

Concerning the resolution of three dimensional inviscid flows using vortex method, most works mainly rely on the 3D formulation exposed here, which is a direct extension of the 2D method, but it has to be noticed that the very first proposition of 3D vortex method was based on a different approach, called the vortex filament method. The latter was introduced by Leonard [30–32] and leans on the theorems of Helmholtz and Kelvin, stating that tubes of vorticity retain their identity and move as material elements in an inviscid flow. In this approach, the quantity carried by the Lagrangian elements is a scalar, as in two dimensions, and the elements are now deformable lines (or filaments) instead of discrete points. Also, as for two-dimensional vortex schemes, the circulation of the particles remains constant in time. In practice, vortex filament methods were used to perform one of the first 3D simulations in CFD by Leonard [31] and were also used more recently by Angelidis and Neyret [33] with smoke simulations in graphic works. However, despite their very natural approach from a physical point of view, they suffer from several weaknesses: first of all, they are most of the time quite difficult to initialize for a given flow. Only specific problems like vorticity rings and jets are spared from this trouble. Also, some filament “surgeries” might become necessary in order to avoid tiny loops or to reconnect neighboring filaments, which make the method complex to use. Eventually, there is no straightforward way to include the physical diffusion effects in the model with filament method.





**Figure 1.** In Vortex blob methods, the vorticity  $\omega^h$  at location  $\mathbf{x}$  is approximated as the sum of the local circulations  $\alpha_p^h$  carried by the neighboring particles of volume  $v_p$  and located in the support of the  $\zeta_\epsilon$  mollified function.

### 2.2. Convergence Issues

What distinguishes vortex methods from grid-based methods is the fact that they are built in such a way that many inviscid flow invariants are conserved [22]. This makes them particularly interesting since they ensure a correct qualitative answer, even in the case of underresolved simulations. Besides this important feature, vortex methods indisputably belong to the category of plain numerical methods by means of their complete numerical analysis and their theoretical proofs of convergence.

The works of Hald in 1979 [34] and Beale and Majda in 1982 [35] provide the complete proof of convergence as part of the vortex blob method in two dimensions. Their analysis included the influence of the numerical parameters involved in the model, namely the blob radius  $\epsilon$  (that is to say the cutoff width) and the particle spacing, denoted  $h$ . It turned out that the vortex blob trajectories tend to the exact one when the number of eddies increases, more precisely when  $h = \mathcal{O}(\epsilon)$ . This result allows us to reach high orders of convergence in two dimensions. An extension of this convergence result to the three dimension case is also given in [35] as well as in the work of Cottet [29], giving birth to new high orders three dimensional vortex methods. The convergence condition  $h = \mathcal{O}(\epsilon)$  is also clearly highlighted numerically by the simulations performed by Knio and Ghoniem [36] concerning the propagation and stability of three-dimensional vortex rings. The temporal discretization is added for the first time in the convergence analysis of the 2D and 3D Euler equations by Anderson and Greengard in 1985 [28]. The most important result which came out of this study is the fact that these methods do not suffer from any Courant-Friedricks-Levy (CFL) condition, which constraints the time step to be related to the minimal grid size in Eulerian methods. Indeed, in vortex methods the stability constraint on the time step only depends on the velocity gradient:

$$\Delta t \leq \frac{C}{\|\nabla \mathbf{u}\|_\infty}, \tag{25}$$

where the constant  $C$ , also called LCFL number (for Lagrangian CFL), must be lower or equal to 1. This condition provides a much less restrictive constraint for the time step compared to purely grid-based schemes.

After this general presentation, one can distinguish several advantages using the discrete vortex method. First, its Lagrangian approach with particle discretization makes it intrinsically close to the flow physics it aims to model and provides a direct scheme to solve the transport of particles and the quantities they carry, based on ODEs systems, without numerical dissipation. Secondly, the vorticity field is a compact field compared to the velocity. It is therefore much more efficient to directly solve and simulate the vorticity field. Then, if the fluid density is constant, the Euler equations for incompressible flows

does not contain pressure term in vorticity formulation. Furthermore, in the discrete vortex method the boundary condition at infinity can be automatically satisfied contrary to most grid-based approaches where external flows must be solved within a finite simulation domain. Finally, the stability condition (25) provided by the vortex method scheme allows larger time steps compared to grid-based methods.

### 3. Viscous Vortex Methods

The first numerical schemes including viscous effects in vortex methods were designed with the desire to conserve the Lagrangian framework. They were consequently purely based on particles. Before starting with the description of the very first viscous vortex method, namely the Random Walk method developed by Chorin in 1973 [37], it is necessary to introduce the concept of viscous splitting of the Navier-Stokes equations. For sake of clarity, in the rest of this paper the approximated quantities  $\omega_\epsilon^h$ ,  $\alpha_\epsilon^h$  and  $\mathbf{u}_\epsilon^h$  will be simply referred to as  $\omega$ ,  $\alpha$  and  $\mathbf{u}$ .

#### 3.1. The Viscous Splitting and Random Walk Method

The original concept of viscous splitting was first introduced by Prandtl, in 1904 [38]. However, its design within the framework of vortex methods was originally proposed in the early 1970s by Chorin [37] in order to develop his Random Walk Method suited to model the diffusive phenomenon. The viscous diffusion equation is:

$$\frac{d\omega}{dt} = \nu \Delta \omega, \tag{26}$$

where  $\nu$  denotes the viscosity of the fluid. This equation is coupled with the Biot-Savart Equation (24) in order to recover the velocity field from the vorticity.

The viscous splitting algorithm consists in performing substeps in which the advective and the diffusive effects are handled separately and successively. Thus, in the vortex method frame, the two steps of the viscous splitting algorithm for a two-dimensional flow are the following:

(1) advection:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p) \tag{27}$$

$$\frac{d\omega_p}{dt} = 0, \tag{28}$$

(2) diffusion:

$$\frac{d\mathbf{x}_p}{dt} = 0 \tag{29}$$

$$\frac{d\omega_p}{dt} = \nu \Delta \omega(\mathbf{x}_p), \tag{30}$$

where in the first step, the particles locations ( $\mathbf{x}_p$ ) are updated using the local flow velocity obtained with the Biot-Savart law and in the second one the vorticity ( $\omega_p$ ) is modified at these new positions by the diffusion. The convergence of the viscous splitting method was proved by Beale and Majda in 1981 [39].

#### Random Walk Method (RW)

The method of Random Walk, created by Chorin [37], consists in solving Equation (30) through the probabilistic interpretation of the Green's function, solution of the diffusion equation:

$$\omega(\mathbf{x}, t) = \int G(\mathbf{x} - \mathbf{y}, \nu t) \omega_0(\mathbf{y}) d\mathbf{y}, \tag{31}$$



which, in  $d$ -dimensions is known to be explicitly equal to:

$$G(\mathbf{x} - \mathbf{y}, \nu t) = \frac{1}{(4\pi\nu\Delta t)^{d/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{4\nu\Delta t}\right). \tag{32}$$

As mentioned previously, the RW method is inherently related to the viscous splitting approach since the vorticity field obtained after the particle push (Equations (27) and (28)) is assigned as the initial condition of the diffusion Equation (30). The diffusion effects are then modeled imposing a Brownian motion to the flow particles whose positions are updated over a time step  $\Delta t$  according to the following scheme:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{u}^n + \mathbf{g}, \tag{33}$$

where  $\mathbf{g}$  is a vector of independent random variables obtained through a Gaussian probability distribution with zero mean and a variance  $\sigma = 2\nu\Delta t$ . Moving the particles in such a way creates a concentration field that converges to the above Green’s function solution of the diffusion equation, as the number of particles tends to infinity. Marchioro and Pulvirenti [40], Goodman [41] and Long [42] successively proved the convergence of the Random Walk method.

Thank to its simplicity, its property to conserve the total circulation and also its capability to easily handle flows around complicated boundaries (see Section 5.1), this method has been extensively used. In terms of applications, we can cite the works of Chorin [37,43], Dommelen [44] and Stansby [45,46] who applied the Random Walk method for the simulations of flows around cylinders. We can also refer to Wang [47] who studied in his thesis several active flow control techniques to prevent the dynamic stall of airfoils and to Lewis [48] who used the Random Walk method to simulate flows over airfoil cascades. However, this viscous vortex method is a stochastic method: it therefore presents several drawbacks. First, it does not exactly conserve the mean position of the vorticity in free space. Next, the computed solutions are noisy due to the statistical errors. Indeed, increasing the viscosity  $\nu$  makes the variance of the random variables larger and consequently deteriorates the solution. Finally, the stochastic feature of the Random Walk method limits its convergence capability. The error is indeed proportional to  $1/\sqrt{N}$  as explained by Milinazzo and Saffman [49], where  $N$  is the number of particles. Therefore, accurate simulations need a very large number of fluid elements, which increases the computational cost and explains their rare application to three-dimensional simulations. One can however cite the use of RW method in the simulation of an impulsively started flow over a 3D cube at  $Re_L = 100$  performed by Gharakhani and Ghoniem [50] (with  $L$  the cube length)

As alternatives to RW, multiple deterministic schemes abounded. The most emblematic ones are described in the following.

### 3.2. Deterministic Viscous Vortex Schemes

#### 3.2.1. Particle Strength Exchange Method (PSE)

Among all the viscous deterministic vortex methods, the PSE, introduced by Degond and Mas-Gallic in 1989 [51] from the previous connected works of Cottet and Mas-Gallic [52,53], is considered as one of the most popular. The mathematical description of the PSE method is here given in a two-dimensional framework but the main ideas of this approach remain the same in three dimensions:

1. Approximate the diffusion operator  $\Delta\omega$  by an integral operator:

$$\Delta\omega(\mathbf{x}) \approx \Delta_\epsilon\omega(\mathbf{x}) = \epsilon^{-2} \int [\omega(\mathbf{y}) - \omega(\mathbf{x})] \eta_\epsilon(\mathbf{y} - \mathbf{x}) \, d\mathbf{y}. \tag{34}$$

where  $\eta_\epsilon$  is a mollified diffusion kernel of order  $r$ , satisfying precise moment properties [22].

2. Substitute (34) in (26):

$$\frac{d\omega}{dt} = \nu \Delta \omega(\mathbf{x}) \approx \nu \varepsilon^{-2} \int [\omega(\mathbf{y}) - \omega(\mathbf{x})] \eta_\varepsilon(\mathbf{y} - \mathbf{x}) d\mathbf{y}. \tag{35}$$

3. The particle approximation of the diffusion equation can be obtained from the numerical integration of (35). The resulting vortex scheme is given by:

$$\omega^h(\mathbf{x}, t) = \sum_p v_p \omega_p^h(t) \delta(\mathbf{x} - \mathbf{x}_p^h), \tag{36}$$

where the particles trajectories  $\mathbf{x}_p^h$  are computed from (27) (based on Biot-Savart law), and  $\omega_p^h(t)$  satisfies the following ODEs:

$$\frac{d\omega_p^h}{dt} = \nu \varepsilon^{-2} \sum_q (v_q \omega_q^h - v_p \omega_p^h) \eta_\varepsilon(\mathbf{x}_q^h - \mathbf{x}_p^h), \tag{37}$$

where the right hand side is obtained from (35) through a discretization of the integral operator taking the particle positions as quadrature points.

Equation (37) can be interpreted as changes in the strength of particles due to the neighboring particles, which explains the name of “Particle Strength Exchange”. Beyond its deterministic framework and its integral formulation, the accuracy and reliability of the PSE method is also based on its capability to conserve the natural invariants of the flow. Moreover, the method is not based on the viscous splitting algorithm, which therefore discards the intrinsic error of the time-stepping process. The method exhibits an error bound of  $\mathcal{O}(\varepsilon^2)$  for simple Gaussian smoothing. However, the quadrature rule used to discretize the integral makes the PSE method strongly dependent on the particles distribution, which has to respect the convergence condition  $h = \mathcal{O}(\varepsilon)$  as well as possible. To circumvent this drawback, particle spatial adaptations often need to be added to the PSE method. This point will be addressed in details in Section 4.

A wide range of applications have been handled so far based on this deterministic approach which may be considered as one of the most efficient and accurate viscous Lagrangian vortex methods. Among them, one must cite the pioneer and reference works of Koumoutsakos and Leonard [54] who applied the PSE method to 2D viscous flows around translating and rotating cylinders for Reynolds numbers ranging from 40 to 9500. One can also cite the works of Ploumhans et al. [55] for the simulation of transient 3D flow past a sphere or those of Yokota et al. [56] for the study of 3D isotropic turbulence in a periodic domain. All these numerical studies highlight the necessity of particle spatial adaption in order to ensure the convergence of the vortex PSE method.

### 3.2.2. Core-Spreading Method (CSM)

This method is not based on integral operators approximating the Laplacian (like in PSE) but directly considers the exact solution of the diffusion Equation (26), given by the Green’s function solution:

$$\omega(\mathbf{x}, t) = \frac{\alpha_p}{(4\pi\nu t)^{d/2}} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_p|^2}{4\nu t}\right) \tag{38}$$

In order to discretize the diffusion equation with a vortex scheme, if one considers the numerical scheme provided by the vortex blob method described in the first section of this paper, one has:

$$\omega_\varepsilon^h(\mathbf{x}) = \sum_p \alpha_{p\varepsilon}^h \zeta_\varepsilon(\mathbf{x} - \mathbf{x}_p^h), \tag{39}$$

where the smooth cutoff function  $\zeta_\varepsilon$  distributes the vorticity on a blob of radius  $\varepsilon$ .

The core-spreading method, first introduced by Kuwahara and Takami in 1973 [57] and later reformulated in 1980 by Leonard [31] has its origin in the observation that, if one chooses  $\zeta_\varepsilon$  as the Gaussian distribution:

$$\zeta_\varepsilon = \frac{1}{(2\pi\varepsilon^2)^{d/2}} \exp\left(-\frac{|\mathbf{x}_q - \mathbf{x}_p|^2}{2\varepsilon^2}\right) \quad (40)$$

then scheme (39) reduces to the solution of the heat Equation (38), provided the variance of the Gaussian blob distribution expands in time according to  $\varepsilon^2 = 2\nu t$ , which explains the denomination of “core-expansion” or “core-spreading” for the method. However, Greengard [58] highlighted that a the straightforward implementation of CSM lacked of consistency and could only give a converged approximation of the Navier-Stokes equations under the very restrictive limit of vanishing viscosity. Indeed, in a general point of view, the expansion in time of the vortex blobs radius is not in agreement with the convergence criterion stating that  $\varepsilon$  must be a small parameter. To circumvent this inconsistency, a spatial adaptation procedure was proposed by Rossi [59] in order to keep control on distance between particles. Based on this improvement (and provided a careful treatment of the local spatial adaptation further described in Section 4.1), the work of Barba et al. [60] demonstrated the means for using CSM in the case of a Lamb vortex dynamics study, and Yokota et al. [56] also used it successfully for their study of 3D isotropic turbulence. This approach conserves the deterministic treatment of the viscous diffusion equation and preserves the meshless framework of vortex methods.

### 3.2.3. Diffusion Velocity Method (DVM)

This method, first introduced by Fronteau and Combis in 1984 [61] and later used in the vortex methods framework by Ogami and Akamatsu in 1991 [62], relies on the idea that the diffusion of vorticity is handled within the advection process by absorbing the diffusion term into the advection one in the Navier-Stokes equations  $\frac{\partial\omega}{\partial t} + \mathbf{u} \cdot \nabla\omega = \nu\Delta\omega$ . This process gives rise to two different velocities: the advection velocity  $\mathbf{u}$  and the diffusion velocity  $\mathbf{u}_d$ . The latter can be considered as an artificial velocity and is defined, in two-dimensions, as follows:

$$\mathbf{u}_d = \frac{\nu}{\omega} \left( \frac{\partial\omega}{\partial x}, \frac{\partial\omega}{\partial y} \right) \quad (41)$$

From this definition, the diffusion problem is now transformed into an advection equation:

$$\frac{\partial\omega}{\partial t} + \nabla(\omega\mathbf{u} + \omega\mathbf{u}_d) = 0 \quad (42)$$

Thus, the transport of vorticity (due to advection and diffusion effects) is modeled by the vortices' movement provoked by the advection velocities and the diffusion velocities. The DVM has been compared to PSE approach in the work of Beaudoin et al. [63] for the 2D simulations of pollutant transport in groundwater, featuring that for such dispersion problem the PSE tends to provide more accurate results while DVM seems to be more stable. However, the DVM requires an important particle overlapping, as precisely highlighted in [63] and in the earlier work of Clarke and Tutty [64] simulating the impulsively started two-dimensional flow past an arbitrary shaped body using DVM associated with a random walk (RW) procedure. Concerning the comparison of diverse vortex diffusion methods one finally refers to a very recent 2D study brought by Lucchesi et al. [65] confronting a direct differentiation approach (non-conservative), with three different PSE schemes and a DVM method (all conservative by construction). An extension to the three-dimensional formulation of the DVM was proposed by Rivoalen et al. [66], where a diffusion vorticity  $\omega_d$  (also called “viscous vorticity”) has to be added to the problem, giving the new vorticity transport equation:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot ((\mathbf{u} + \mathbf{u}_d) \otimes \boldsymbol{\omega}) - (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} - \boldsymbol{\omega}_d \times \boldsymbol{\omega} = 0 \tag{43}$$

which, in the Lagrangian framework, is solved according to the following system of ODEs:

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_p + \mathbf{u}_d \tag{44}$$

$$\frac{d\boldsymbol{\omega}_p}{dt} = (\boldsymbol{\omega}_p \cdot \nabla) \mathbf{u}_p - \boldsymbol{\omega}_d \times \boldsymbol{\omega}_p \tag{45}$$

However, it is important to notice that  $\boldsymbol{\omega}_d$  needs the approximation of the Laplacian operator, which in practice in [66] is performed using a discretization of the integral approximation (as done in PSE method).

These three deterministic viscous methods, namely the PSE, the CSM and the DVM, represent the grid-free viscous approaches that allowed conservative and converged numerical simulations for two and three-dimensional problems. It is interesting however to mention several other methods that were proposed to solve the viscous diffusion equation with vortex schemes, despite their failure in successfully applying it to the three-dimensional case. One can for instance cite the one introduced by Fishelov in 1990 [67], called the Convolution of Cut-off Function method (CCF). It allows us to obtain the Laplacian operator by using explicitly second derivatives of the cutoff function, i.e.,  $\Delta \omega \approx \omega \star \Delta \zeta_\epsilon$ , in the vortex blob scheme:

$$\frac{d\omega_p^h}{dt} = \nu \sum_q (v_q \omega_q^h) \Delta \zeta_\epsilon(\mathbf{x}_q^h - \mathbf{x}_p^h). \tag{46}$$

But this method encounters several weaknesses. It has been first observed to be more sensitive than PSE to particle field distortion, but overall, the formulation (46) lacks the correction term  $-\omega_p \sum_q v_p \Delta \zeta_\epsilon(\mathbf{x}_q^h - \mathbf{x}_p^h)$  and is therefore not conservative [22]. These reasons explain why this approach failed in giving rise to deeper research works.

Among the other vortex schemes proposed to solve the diffusion equation, one can also mention the Vortex Redistribution Method (VRM) of Shankar and van Dommelen (1996) [68]. This method was thought as an alternative to PSE with no need to perform particle rezoning. As PSE, the modeling of diffusion is built on the exchange of circulation between neighbor particles, but in VRM the scheme formulation is not based on integral operators. The exchange of circulation between particles is modeled through the computation of functions that will determine the amount of circulation each particle has to redistribute to its neighbors, located in the area limited by the typical diffusion distance. The definition of the redistribution fractions is designed such that local circulation as well as linear and angular momentum are conserved. However, their evaluation implies many parameters whose computation might be very costly (even more than the cost of velocity computation through Biot-Savart law [22]), which in practice makes this approach difficult to use.

For an exhaustive description about grid-free viscous vortex methods, one can refer the reader to the detailed literature survey given in Barba’s thesis [69] and to Cottet and Koumoutsakos [22].

The important number of diverse viscous vortex schemes illustrates the difficulty of modeling viscous incompressible flows with a grid free vortex method. As emphasized in the present section, the main limitation of all the proposed schemes does not come from their capability of modeling the viscous effects themselves, but from the necessity to ensure particle overlapping throughout the whole calculation. Studies on spatial adaptation procedures were therefore developed in parallel in order to address this key point. We describe them in the following section.

#### 4. Spatial Adaptation and Hybrid Vortex Methods

In pure Lagrangian method, the particle distribution may undergo distortion, that manifests itself by the clustering or spreading of the flow elements in high strain regions.

This accumulation or dispersion phenomenon is then translated in inaccurate computations because of a deteriorated communication between particles. As meaningful illustrations of the distortion effect one can cite the test case of the axisymmetric inviscid vortex patch, illustrated for instance by Rossinelli and Koumoutsakos [70]. In this example, one can observe the distortion of the initial and uniform particle distribution due to the important shear. One can also see in Henshaw et al. and in Krasny's studies [71,72] comparable behaviors concerning the appearance of non-physical small scales in vortex sheet calculations in the context of two dimensional inviscid flows. Therefore, and as mentioned in Section 2.2, the keypoint of vortex methods rely on the necessity of particle overlapping, that is to say, the need for the cutoff size  $\varepsilon$  to be at least of the order of  $h$ , the distance between particles.

Several solutions have been designed in order to overcome the distortion problem. One can separate these methods into two main classes.

#### 4.1. The Meshless Rezoning Methods

The first class deals with methods involving a change in particle strength in order to correct the vorticity field such that the particle distribution better approximates the vorticity field. Among these methods one can first refer to the approach developed by Beale in 1988 [73] which consists in correcting the circulation at each time step through an iterative procedure. This technique has been successfully used by Chorin [74] to model incompressible viscous flow with a PSE scheme. However, the iterative process may be slow to converge in case of particle clustering or irregularities in the vorticity field. Both Beale and Chorin defined this circulation evaluation method as a problem of "scattered data interpolation".

In this spirit, Barba et al. [60] later proposed in 2005 a scattered data interpolation using Radial Basis Functions (RBF) as a rezoning approach in the context of vortex method. In this technique, as the vorticity  $\omega$  is known only over a finite number of positions  $\mathbf{x}_p$ , the objective is to find an approximation to the assumed piecewise continuous function  $\omega(\mathbf{x})$  as follows:  $\omega(\mathbf{x}) = \sum_{p=1}^N a_p \phi_p(\mathbf{x} - \mathbf{x}_p)$  where  $a_p$  denote the coefficients that have to be determined by collocation using the known values of  $\omega$  at  $\mathbf{x}_p$ , and where the basis functions  $\phi_p$  have compact support and are often chosen to be univariate and only dependent on the distance between two particles (radial functions). The solution of such interpolation requires to solve a  $N$ -linear equation (one refers to [69] for a detailed description of the algorithmic issues about such approach). The RBF interpolation is a grid-free formulation and the new particles may be located in any spatial arrangement.

Another type of meshless rezoning techniques are the Free-Lagrange methods. The latter were initially designed to model the diffusion phenomenon but they actually offer a good framework to handle distortion problems. They consist in creating a triangulation, whose nodes are the particles locations, which is then used to interpolate the circulation carried by the flow elements. For instance, the Delaunay triangulation, suggested by Russo and Strain in 1994 [75] and obtained connecting the particles when they belong to adjacent polygons in a Voronoi diagram, enables to successfully avoid the distortion of the particle distribution and provides a second-order accurate vortex scheme. Another example among these Free-Lagrange methods is the adaptive quadrature method introduced by Strain in 1997 [76]. This approach allows us to increase the order of accuracy of the usual triangulation methods by considering sub-elements that would combine several triangles together.

One can also cite the strategy introduced by Schrader et al. in 2010 [77] which can be considered as a generalization of the PSE method. It indeed proposes to correct the method given by Eldredge et al. in 2002 [78] consisting in approximating the derivatives on scattered particles locations. This corrected framework of the PSE scheme enables one to achieve the convergence without any overlapping condition and regardless of the resolution. However, the convergence of the latter was limited to a certain range of resolutions.

Finally, we can refer to the recent approach proposed by Rossi et al. in 2015 [79], called the Regular Point Distributions (RPD). This technique, introduced within the Diffused Vortex Hydrodynamics method (DVH) designed by the same authors, consists in defining an appropriated set of points on which are generated the new particles after each diffusion steps. This set of points is generated with a “package” algorithm which, in the presence of an immersed obstacle, creates points regularly around the body (of any geometry) in order to accurately prescribe the no-slip boundary condition at the surface (the enforcement of boundary conditions in vortex methods will be addressed in details in the next main section).

#### 4.2. The Remeshing Methods

The second class of methods designed in order to remedy the distortion problem are based on a frequent physical re-location of the particles onto new positions where the overlapping condition is satisfied. These methods allow us to directly control the distance between the particles, thus ensuring the convergence of the solution. The following of this section is dedicated to the description of the re-location methods principally used by now, based on the so called “remeshing” technique.

From the pioneer works of Huberson and Jollès (1990) [80] and Koumoutsakos and Leonard (1995) [54], the remeshing approach (also called “redistributing” or “regridding”) has been widely and is still successfully used in vortex methods. It consists in redistributing the particles on an underlying Cartesian grid. The description of the regridding process is here proposed for the one-dimensional case. Let us denote by  $\tilde{x}_q$  the position of the particle  $q$  carrying the circulation  $\tilde{\Gamma}_q$  (or vorticity  $\tilde{\omega}_q$ ). These particles are remeshed at the grid points  $x_k$  of a uniform Cartesian grid, with spacing  $h$ , using the interpolation kernel  $W$  of support  $[-M_s, +M_s]$  like follows:

$$\Gamma_k = \sum_q \tilde{\Gamma}_q W\left(\frac{x_k - \tilde{x}_q}{h}\right). \tag{47}$$

In other words, this procedure means that every grid point  $k$  receives the sum of the weighted values of the circulation carried by the particles for which  $k$  is included in the interpolation support.

Let us recall that one of the strong advantages of the vortex methods lies in their capability to satisfy the conservation properties. In order to preserve this important feature, the interpolation kernel has to be constructed such that the successive moments are conserved. For instance, to satisfy the total circulation invariance (and more generally, the conservation of mass),  $W$  must ensure:

$$\sum_k W\left(\frac{x - x_k}{h}\right) = 1. \tag{48}$$

The conservation of the first  $p$  moments is given by the following conditions:

$$\sum_q x_q^\alpha W\left(\frac{x_q - x}{h}\right) = x^\alpha, \quad 0 \leq |\alpha| \leq p - 1, \tag{49}$$

and  $W$  is therefore an interpolation kernel of order  $p$  (we refer to [22] for the proof of this assessment). For  $p = 0, 1$  and  $2$ , the conservation of the total circulation, the linear impulse and the angular impulse are cumulatively ensured.

In the following, we will describe the different families of remeshing kernels  $W$  by giving their construction as well as their properties.



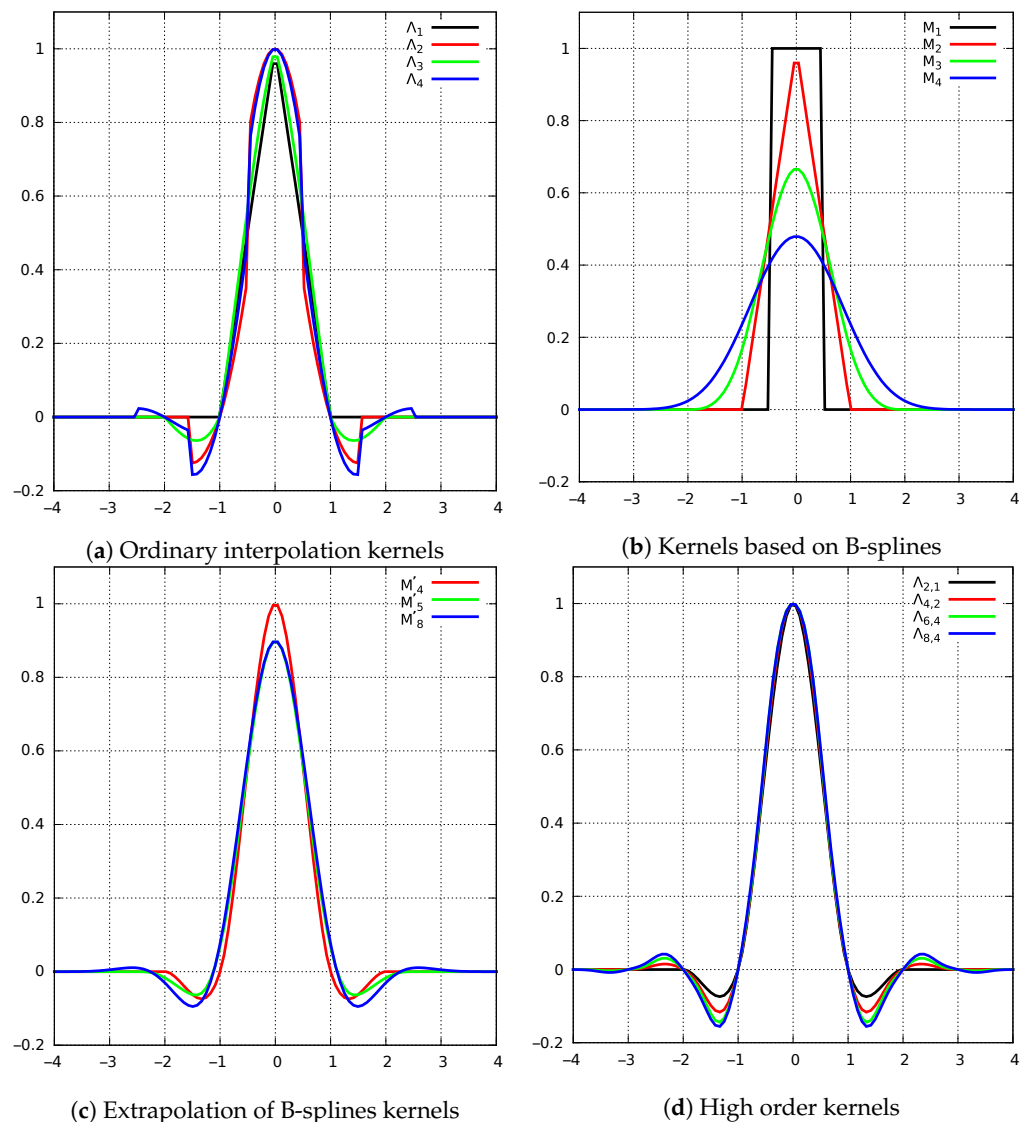
### 4.2.1. Remeshing Kernels

#### The Ordinary Interpolation Kernels

This group includes the kernels whose construction is only based on the satisfaction of conditions (49). These kernels, denoted  $\Lambda_p$  where  $p$  refers to the maximum order of conserved moments, are piecewise polynomial functions of degree  $p$ . Their construction requires the calculation of  $p(p + 1)$  coefficients, obtained from the relations (49). For  $p = 1$ , one gets the linear interpolation kernel, satisfying mass conservation:

$$\Lambda_1(x) = \begin{cases} 1 - |x| & \text{if } |x| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (50)$$

As can be seen on Figure 2a where are also plotted  $\Lambda_2, \Lambda_3$  and  $\Lambda_4$ , this type of kernels suffer from a lack of regularity. One can notice that  $\Lambda_2$  is not even continuous, which is expected to lead to inaccurate solutions. In order to reduce the creation of oscillations due to the poor regularity of the kernels, Magni and Cottet [81,82] proposed to insert correction formulas in the expression of  $\Lambda_p$ .



**Figure 2.** Graphical representation of the different families of remeshing kernels. For each family, several kernels of different orders are plotted (these plots have been constructed from the equations given in Section 4.2.1).



### The B-Splines-Based Kernels

In 1946, Schoenberg [83] introduced smooth kernels, denoted  $M_p$ , obtained using B-splines :

$$M_p(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left( \frac{\sin \xi/2}{\xi/2} \right)^p e^{ix\xi} d\xi \tag{51}$$

For  $p = 1$  one gets the top-hat function,  $M_1$ . The  $p$ -order kernels  $M_p$  thus correspond to the successive convolutions of  $M_1$ :  $M_p = M_1^{(*p)}$  and each kernel  $M_p$  is of class  $C^{p-2}$  (Figure 2b).  $M_3$  and  $M_5$  particularly distinguish themselves for their non-oscillatory properties, as illustrated by Magni [84]. Nevertheless, the  $M_p$  kernels only ensure the conservation of the two first moments, which strongly limits the accuracy of the remeshing procedure.

### Extrapolation of B-Splines-Based Kernels

In order to achieve higher order remeshing schemes while preserving their smoothness properties, Monaghan [85] proposed in 1985 an extrapolation of the  $M_p$  kernels. It consists in deriving linear combinations of the  $M_p$  kernels and their derivatives, more precisely  $W = \sum_{l=0}^{p/2} \alpha_{l+1} x^l M_j^{(l)}$ , such that the successive continuous moments of  $W$  are canceled:

$$\int y^\alpha W(y) dy = \begin{cases} 1 & \text{if } \alpha = 0 \\ 0 & \text{if } 1 \leq \alpha \leq p. \end{cases} \tag{52}$$

With this approach, Monaghan obtained the so-called  $M'_4$  kernel defined by:

$$W = M'_4 = \frac{1}{2} \left( 3M_4(x) + x \frac{dM_4}{dx}(x) \right) \tag{53}$$

$$= \begin{cases} 1 - \frac{5}{2}|x|^2 + \frac{3}{2}|x|^3 & \text{if } |x| \leq 1 \\ \frac{1}{2}(1 - |x|)(2 - |x|)^2 & \text{if } 1 \leq |x| \leq 2 \\ 0 & \text{otherwise.} \end{cases} \tag{54}$$

This kernel is of class  $C^1$ , it is piecewise cubic, and its support is of size 4 on  $[-2, 2]$ . Locally this kernel offers a third-order accurate scheme (although its global order is 2) and conserves the first three invariants when used in vortex methods (total circulation, linear impulse and angular impulse). In practice, the  $M'_4$  is a very good compromise between regularity, accuracy and computational cost, which explains that it is still widely used in vortex methods [6,8,86]. However, these extrapolated smooth kernels do not systematically satisfy the following interpolation property:

$$W(i) = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{55}$$

For instance, the  $M'_4$  kernel does satisfy this condition, but it is not the case of the one derived from  $M_8$  (Figure 2c). Nevertheless, this property is essential since it dictates the algebraic conservation of the exact solution if the velocity is zero (that is to say if the particles do not move from their original location on the grid).

### High Order Kernels

Much later, Bergdorf and Koumoutsakos [87] proposed in 2006 a higher order kernel, called  $M'_6$  and in 2014, Cottet et al. [88] used high order kernels in a systematic way. These high order kernels satisfy simultaneously the interpolation property, the smoothness and the moment conditions to any desired order. Their construction relies on the satisfaction of the five following properties:

- P1:  $W$  has support in  $[-M_s, +M_s]$ ,
- P2:  $W$  is even and piecewise polynomial of degree  $M$  in intervals of the form  $[i, i + 1]$ ,
- P3:  $W$  is of class  $C^r$ ,
- P4:  $W$  satisfies the moment properties (49) for a given value of  $p$
- P5:  $W$  satisfies the interpolation property (55).

In [88], a new global notation is proposed for these high order remeshing kernels and stands as  $\Lambda_{p,r}$  where  $p$  denotes the order to which moment conditions are satisfied and  $r$  refers to the regularity of the kernel. Figure 2d depicts some of these kernels. In particular, the kernels  $\Lambda_{2,1}$  and  $\Lambda_{4,2}$  respectively correspond to  $M'_4$  and  $M'_6$ .

#### 4.2.2. The Directional Splitting

For multi-dimensions problems, the remeshing step is classically performed through a tensorial product of the chosen 1D kernel formula. For example, if the desired kernel has a 1D-support of  $S = 2M_s$  points, then the number of operations needed to interpolate one particle on the grid is proportional to  $S^2$  for a two-dimensional problem and to  $S^3$  for a three-dimensional one. A new method was proposed in 2012 by Magni and Cottet [82] for the particle redistribution procedure. It consists in successively solving monodirectional sub-problems, including the advection and the remeshing step. In other words for a  $N$ -dimensional problem, one applies the following algorithm for each particle:

```

for d ∈ {1, ..., N}:
    advection in direction d
    remeshing in direction d
    
```

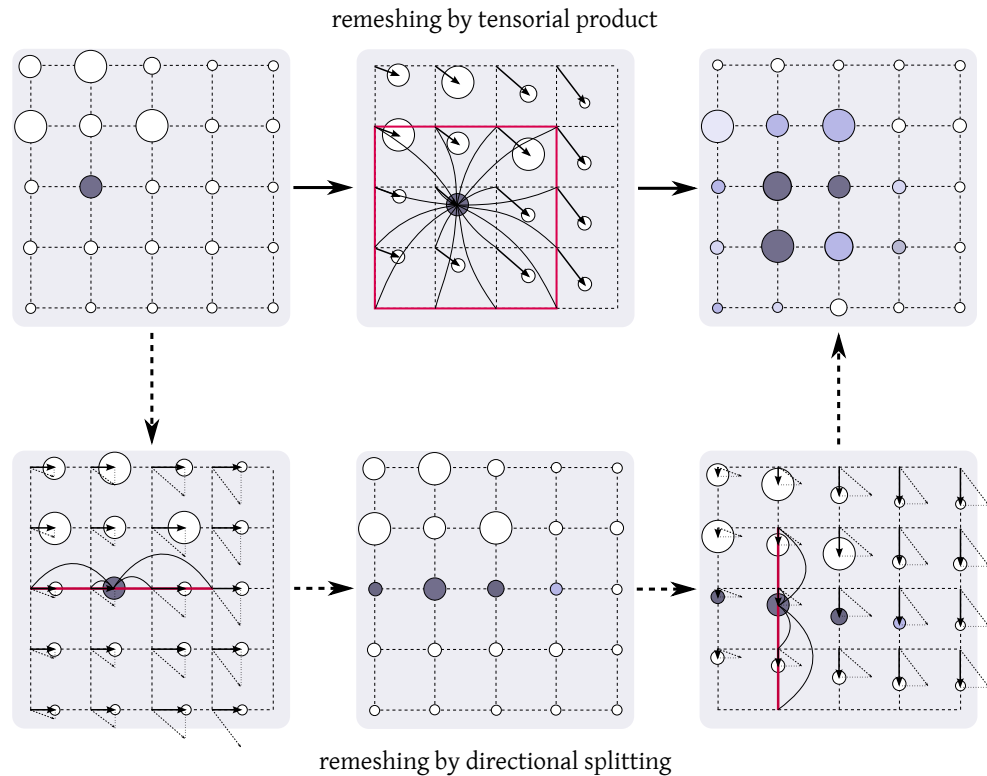
We refer to Figure 3 for a comparative pattern of tensorial product and directional splitting methods. As a consequence, if the chosen kernel contains  $S$  points in its 1D-support, the number of operations with the directional splitting method goes from  $\mathcal{O}(S^2)$  to  $\mathcal{O}(2S)$  in 2D and from  $\mathcal{O}(S^3)$  to  $\mathcal{O}(3S)$  in 3D. The directional splitting consequently allows for a drastic reduction of the computational cost in terms of regridding operations. The full convergence proof of the 1D advection-remeshing scheme, considering a smooth non-constant velocity and remeshing kernels of type  $\Lambda_{p,r}$ , was brought by Cottet et al. in 2014 [88], thus providing a robust and theoretical numerical framework to the directional splitting approach.

The increase of the time-order of the directional splitting scheme (limited to 1 in the form described above) was proposed by Magni and Cottet [82] through the so called Strang splitting scheme, which was successfully used by Cottet et al. [88] in the context of three-dimensional passive scalar advection problems. It consists in performing the directional advection/remeshing with half time steps, except for the last direction where one can gather the operations on  $[t_n; t_{n+1/2}]$  and  $[t_{n+1/2}; t_{n+1}]$  together. The Strang splitting scheme is therefore of order 2 in time. The different substeps of this approach are given below for the 2D case (the same principle applies for the 3D case):

- advection + remeshing in the X-direction on  $[t_n; t_{n+1/2}]$
- advection + remeshing in the Y-direction on  $[t_n; t_{n+1}]$
- advection + remeshing in the X-direction on  $[t_{n+1/2}; t_{n+1}]$

With the Strang splitting method, a multidimensional remeshing thus costs  $\mathcal{O}(3S)$  (respectively  $\mathcal{O}(5S)$ ) operations per particle in 2D (respectively in 3D).

The introduction of an underlying grid in order to ensure particle overlapping modifies the essence of the original vortex methods and rapidly gave birth to the so-called “hybrid” vortex methods. The last part of this section is dedicated to the description of the two main families of hybrid vortex methods, namely the Domain Decomposition method on one side and the Vortex-Particle-Mesh (VPM) methods on the other side, with the Vortex-In-Cell (VIC) and the semi-Lagrangian vortex methods among the latter.



**Figure 3.** Remeshing steps using a tensorial product (on top, depicted by plain arrows) and a directional splitting [82] (on bottom, depicted by dashed arrows). The red lines indicate the support of the remeshing kernel. In this example, the kernel has a 1D-support of 4 points.

### 4.3. Hybrid Vortex Methods

Hybrid vortex methods were developed in order to overcome the main weaknesses of the pure Lagrangian vortex methods which essentially lie in the treatment of the viscous effects (see Section 3) and, as we will see in the next section, in the near wall boundary conditions. In order to preserve the robustness of the global scheme, characterized by minimal numerical dissipation, these methods still involve a Lagrangian framework for the advection process but they also handle a fixed grid. The presence of this grid facilitates the prescription of the no-slip boundary conditions as well as the modeling of the diffusive term, using Eulerian schemes (like finite-differences, finite-volumes, finite-elements or spectral methods) while ensuring the particle overlapping condition. In the following we will consider a vortex method as *hybrid* whenever the resolution of the vorticity transport equation and the velocity Poisson equation will be handled both on the particles and on an underlying grid.

Before introducing some hybrid vortex methods, we need to define the common thread between all of them which stands in the way to exchange information between Eulerian and Lagrangian schemes. More precisely, hybrid methods require us, on one hand, to transfer the values of the flow fields  $\mathbf{a}$  from the particles to the grid points (we will denote it the *P2M* process, for *Particle to Mesh*) and, on the other hand, to transfer the values of  $\mathbf{a}$  from mesh points to the particle distribution (we will denote it the *M2P* process, for *Mesh to Particle*). It is important to note that the *P2M* and *M2P* processes do not imply a physical particle relocation, they materialize an exchange of information.

Let us first define the *P2M* scheme, also called the assignment scheme. It is defined by the following scheme:

$$\mathbf{a}_k = \frac{1}{\mathbf{V}_k} \sum_p v_p \mathbf{a}_p \phi_k(\mathbf{x}_p), \tag{56}$$

where  $V_k$  corresponds to the volume surrounding the grid point  $x_k$  (this volume equals to  $h^3$  if the mesh is Cartesian with a uniform mesh size  $h$ ) and the basis function  $\phi_k$  is defined by:

$$\phi_k(x_p) = \phi\left(\frac{x_p - x_k}{h}\right), \tag{57}$$

and satisfies:

$$\sum_k \phi_k \equiv 1. \tag{58}$$

If the considered grid is Cartesian, the basis function  $\phi$  may be taken as one of the  $M_p$ ,  $M'_p$  or  $\Lambda_{r,p}$  functions described in the previous subsection. In that case, the remeshing procedure presented previously may be considered as a particular case of the  $P2M$ .

On the other hand, the  $M2P$  scheme, also called the interpolation scheme, yields:

$$a_p = \sum_k a_k \phi_k(x_p) \tag{59}$$

These  $P2M$  and  $M2P$  formula lead to conservative schemes. They are depicted in Figure 4.

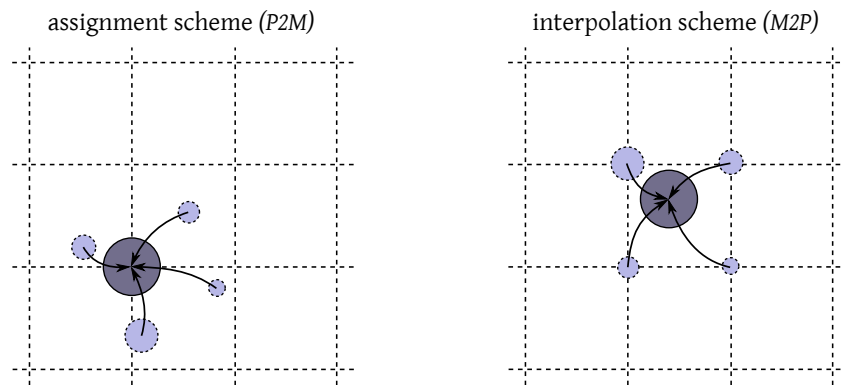
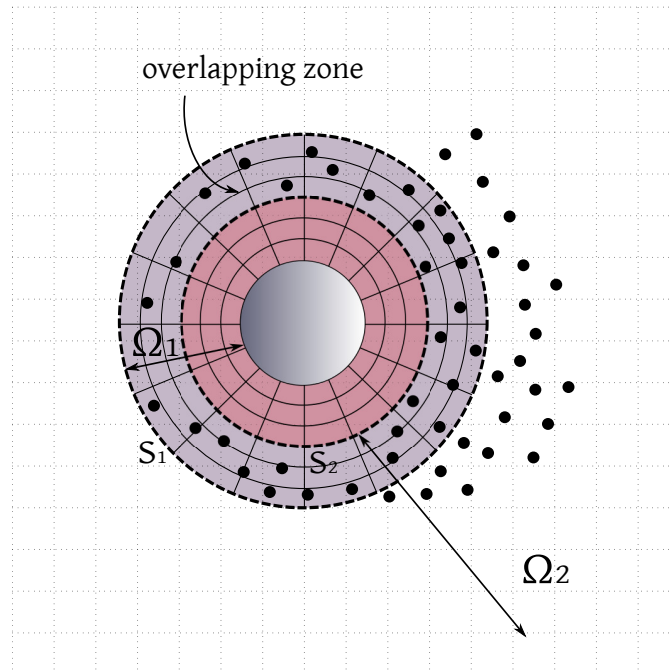


Figure 4. Assignment and interpolation schemes.

#### 4.3.1. The Eulerian-Lagrangian Domain Decomposition Method

The first type of hybrid vortex approach presented in this section is the Eulerian-Lagrangian domain decomposition method, which was proposed by Cottet in 1991 [89] in the context of inviscid flows. The main idea relies on the fact that grid methods and Lagrangian methods are used in different parts of the computational domain. In the flow regions close to immersed obstacles, the use of grid-based schemes enables one to accurately impose the boundary conditions and on the other hand the grid-free vortex method, which is implemented elsewhere, ensures an accurate modeling of the advection-diffusion effects and also provides a simplified treatment of the domain boundary conditions. In the same spirit, Cottet et al. [90] considered an overlapping zone between the Eulerian (grid-based) domain  $\Omega_1$  and the Lagrangian (grid-free) domain  $\Omega_2$  (Figure 5), implying a Schwarz method in the overall algorithm. It is important to note that the remeshing procedure performed in the Lagrangian domain  $\Omega_2$  is not as crucial in this context as in a pure Lagrangian framework (cf. Section 3). Indeed, the distortion effects mainly occur in the vicinity of the wall boundaries, where the highest velocity gradients are observed.



**Figure 5.** Domain decomposition for flow past a circular obstacle with overlapping Eulerian  $\Omega_1$  and Lagrangian  $\Omega_2$  domains. In this case, inspired from Cottet et al. [90],  $\Omega_1$  is defined by a polar mesh and a uniform Cartesian grid lies in  $\Omega_2$  for particle remeshing.

Huberson and Voutsinas [91] also report simulations of flows past circular cylinders realized with this Eulerian/Lagrangian decomposition method, as well as the example of a wind turbine operating in a non-axial wind flow, where a finite-differences scheme is used in the near wake region. One can cite as well the works of Guermond et al. who considered matching domains (no overlapping) [92], which makes the definition of interface boundary conditions more difficult but enables a reduction of computational operations.

#### 4.3.2. From Vortex-In-Cell (VIC) to Semi-Lagrangian Vortex Methods

The Vortex-In-Cell is a hybrid vortex method which was proposed in the pioneer studies of Christiansen in 1973 [93]. It relies on dealing with an underlying Cartesian grid and using either Eulerian schemes (on the grid) or Lagrangian vortex schemes (on particles), according to their respective strengths, to solve the different parts of the governing equations. This method is based on a splitting algorithm (inspired from the Chorin’s viscous splitting algorithm described in Section 3.1), also called a fractional step algorithm or an operator splitting algorithm. Concretely, in a VIC method, the advection is performed by updating the positions and the vorticity of the Lagrangian particles through a set of ODEs. On the other hand, Eulerian schemes are handled on the underlying grid to efficiently compute the particle velocities and to solve the diffusion terms for which pure vortex method encounter difficulties, as widely discussed in Section 3. For three dimensional flows, the stretching term is often resolved on the mesh for the sake of simplicity.

The general Vortex-In-Cell fractional step algorithm can be described as follows:

##### 1. Poisson equation:

- (a) Assign particle vorticity values to the grid using a *P2M* formula.
- (b) Compute the velocity field solving the following equation on the grid:

$$\Delta \mathbf{u} = -\nabla \times \boldsymbol{\omega}. \tag{60}$$

## 2. Advection:

- (a) Interpolate the velocity field on the particles using a *M2P* formula.
- (b) Perform a Lagrangian advection of the particles and get their new positions and vorticity:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \quad \frac{d\boldsymbol{\omega}}{dt} = 0 \quad (61)$$

- (c) Assign vorticity values and element volumes to the grid using a *P2M* formula.

## 3. Stretching (in three-dimensions only):

- (a) Solve the stretching equation on the grid by differentiation of the velocity field:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} \quad (62)$$

Note: One can also write the stretching equation in its conservative form, namely

$$\partial \boldsymbol{\omega} / \partial t = \text{div}(\boldsymbol{\omega} : \mathbf{u}) \equiv (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} - \mathbf{u} \text{div}(\boldsymbol{\omega})$$

## 4. Diffusion:

- (a) Solve the diffusion equation on the grid and get the final grid values for the vorticity field:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nu \Delta \boldsymbol{\omega} \quad (63)$$

- (b) Interpolate the particle values from the grid using a *M2P* formula.

As a typical example of study based on this type of VIC algorithm one can cite the work of Mortazavi and Giovannini [94]. In this study, dedicated to the simulation of 2D flow past thin and thick plates at  $Re_H = 7000$  (with  $H$  the thickness of the plate), the Poisson equation (step 1) is solved with a finite element method (on a mesh), the advection (step 2) is solved with a usual vortex blob approach (where the core diameter is taken equal to the mesh size) and the diffusion (step 4) is realized through a Random Walk method. Besides, let us precise that the algorithm structure written above corresponds to the original version of the VIC algorithm, following the idea of Christiansen. We note that this initial version actually suffers from the same difficulties as the one experienced by purely Lagrangian approaches because of the distortion effects provoked by the Lagrangian resolution of the advection equation (step 2). The emergence of the remeshing technique in the mid 1990s allowed us to overcome this problem and the VIC method, also known nowadays as the “Vortex Particle-Mesh method” (VPM) (or “Remeshed Vortex methods”) then became particularly attractive and efficient to handle very different types of applications [95–98]. One can find a wide range of differences among all the existing Vortex Particle-Mesh methods, which can be explained by the very modular structure of the fractional step algorithm they are based on. Concerning the resolution of steps 3 and 4, a common trend is to use a finite-differences scheme (like in the four references given above [95–98]), while the Poisson equation, in step 1, is often handled with a Fast Fourier Transform method or a Fast Multiple Method (FMM) (these algorithm aspects will be explained more precisely in Section 6.1). The numerical order of a VIC method therefore depends on the order of each scheme adopted in each substep of the operator splitting algorithm. An important difference between the diverse existing VIC schemes concerns the remeshing procedure in step 2. For example, in Caprace et al. [99], the particles are remeshed on the underlying Cartesian grid only every 5 time steps, while in Mimeau et al. [100] the remeshing procedure relocate the particles on the grid points every time steps, implying the presence of a particle at each node of the Cartesian mesh, at any time. In this particular latter case, one talks about “Semi-Lagrangian Vortex methods”.

Today, the Vortex Particle-Mesh methods may be considered as one of the most commonly used types of hybrid vortex methods thanks to their relevant advantages we can recapitulate in the following list:



- Possibility to use Fast Poisson solvers for the velocity computation instead of direct summations using the Biot-Savart integral,
- Easy treatment of the viscous effects using finite-differences schemes on the grid (or finite-volumes schemes, spectral schemes) for the resolution of the diffusion equation,
- Easiness of implementation compared to domain decomposition methods,
- Basis for mixing VM with other methods, easy model coupling,
- Prevalence of the less restrictive Lagrangian CFL condition (Equation (25)) for the whole stability,
- Satisfaction of the particle overlapping condition all along the simulation through remeshing procedures, whose computational cost may be greatly decreased by the use of directional splitting schemes (see Section 4.2.2),
- Limitation of the diffusive effects in the advection step thanks to the Lagrangian framework (provided the use of regular and high order remeshing kernels like  $M_4^4$  or more generally  $\Lambda_{p,r}$ ).

For all these reasons, the VPM methods (i.e., VIC, semi-Lagrangian) have been extensively used in DNS (Direct Numerical Simulations) to discretize the three-dimensional Navier-Stokes equations, especially in the case of highly transitional regimes with Reynolds numbers ranging from 1000 to 10,000. Among the large quantity of CFD studies realized with VPMs, one can mention for example the work of van Rees et al. [97] who carried out the 3D Taylor-Green vortex case at  $Re_\Gamma = 1/\nu = 1600$  and successively compared their results to the one obtained with spectral methods, as well as the studies of Morgenthal and Walther [101] or Kudela and Kozłowski [102] where a VIC method is used to model incompressible flows past cylinders up to  $Re_D = 3000$  and  $Re_D = 9500$  respectively (with  $D$  the cylinder diameter). One can moreover refer to the work of Cottet and Poncet [103] which proposed one of the first 3D flow control studies using VIC, where an optimal rotation of a 3D cylinder is designed in order to reduce the drag force. In terms of flow control one also notices the investigations of Creusé et al., based on a VIC method, concerning the active control of a flow over a backward-facing step using pulsing jets [104]. We refer the reader to Section 7 for further applications handled with VIC and semi-Lagrangian vortex methods that clearly highlight the above listed advantages of these methods.

Another type of hybrid vortex method one can also cite to conclude this section is the one designed recently by Kornev in 2018 [105] where the mesh-free simulation is embedded into the grid based one. It consists in representing the large scale field on the grid, whereas the small scale field is evaluated through a pure Lagrangian vortex method, the separation of the large scale vortices from the small ones being based on a LES (Large Eddy Simulation) filtering procedure. By far, this method has only been tested on simple two dimensional canonical flows and its application to more complex physics remains to prove.

## 5. Treatment of the Boundary Conditions

The ability of correctly handle boundary conditions in a very important issue for any numerical method in CFD. As is well known, the boundaries and the flow viscosity are the only source of vorticity, which explains the efforts made in order to design a method able to accurately model the flow behavior in this region, essentially governed by the no-slip boundary condition. In the context of vortex methods, the major difficulty lies in the fact that the no-slip boundary condition only involves the velocity field in its expression whereas these methods are based on a vorticity formulation of the Navier-Stokes equations. In this section we will give a global description of the main approaches dealing with boundary conditions in vortex methods. We will therefore consider the velocity-vorticity formulation of the Navier-Stokes equations and we will restrict this survey to solid and rigid obstacles.

In the context of Lagrangian vortex methods, the objective is to express the boundary conditions in terms of the vorticity field, equivalent to the no-slip boundary condition  $\mathbf{u}(\mathbf{x}_s) = \mathbf{u}_s$ , where  $\mathbf{u}_s$  denotes the velocity of the solid body and  $\mathbf{x}_s$  any location on the solid



surface. To write these boundary conditions one may choose a Dirichlet formulation (wall vorticity) or a Neumann formulation (wall-normal vorticity flux). The Neumann form is often preferred since it directly governs the local production of vorticity at a solid wall. Indeed, the wall-normal vorticity flux, given by  $\frac{\partial \omega}{\partial \mathbf{n}}$ , measures the vorticity that enters the flow from the boundaries. This flux is actually responsible for the total production of vorticity at the no-slip wall.

The approaches designed to model the boundary conditions can be divided into three main categories, exposed hereafter. The first family of methods belongs to pure Lagrangian methods; it is based on viscous splitting and mimics the vortex creation at the interface through the generation of vortex sheets. The two other families involve the presence of a mesh or a lattice and are therefore often used within Vortex-Particle-Mesh methods. The first one, known as the Boundary Elements Method (BEM) enforces the no-slip by defining vorticity flux boundary conditions on panels located at the wall, and the second one prescribes the boundary conditions by using Immersed Boundary Methods (IBM).

### 5.1. Boundary Treatment Based on Viscous Splitting and Vorticity Creation

The first successful approach designed to handle the boundary conditions in vortex methods for fluid flows is due to Chorin in 1973 [37], later improved by the same author in 1978 [106]. The method he introduced, called the vortex sheet/vortex blob method, uses the viscous splitting technique (described previously in Section 3.1) and relies on creating a vortex sheet at the solid boundary in order to vanish the slip velocity and thus to enforce the required no-slip boundary condition. Practically, the wall is discretized into segments. Each of these segments is associated to a local slip velocity where vortex elements are created at each time step to cancel it. We detail the different steps of the algorithm used in the improved method proposed in [106] to discretize the Navier-Stokes equations, with boundary conditions. More precisely, this approach is based on the coupling of the Prandtl boundary-layer equations near the surface with the Navier-Stokes equations everywhere else. The Prandtl equations are expressed in a coordinate system where  $x$  and  $y$  respectively denote the tangential and the normal coordinates, and where  $(u, v)$  corresponds to the velocity vector field in this system. The solid wall is assumed to be at  $y = 0$  and the fluid domain is defined by the half-space  $y \geq 0$ . In the boundary region, the Prandtl boundary-layer equations rely on the assumption that vorticity diffusion mainly occurs in the direction normal to the wall and high velocity gradients are essentially along the component tangential to the wall (see Figure 6); they therefore can be written as:

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \nu \frac{\partial^2 \omega}{\partial y^2} \quad (64)$$

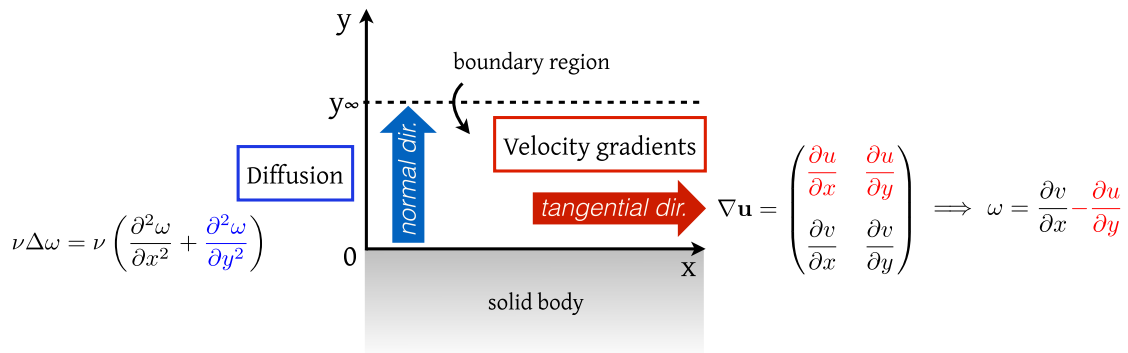
$$\omega = -\frac{\partial u}{\partial y} \quad (65)$$

They are completed by the continuity equation and the following boundary conditions:

$$\mathbf{u} = 0 \text{ at } y = 0, \quad (66)$$

$$u(x, y_\infty) = u_\infty(x), \quad (67)$$

where Equation (66) imposes the no-slip boundary condition at  $y = 0$  and Equation (67) expresses a far-field condition for the  $u$  component when  $y \geq y_\infty$ . The value  $y_\infty$  is often taken of the order of the boundary layer thickness.



**Figure 6.** Graphical representation of the main directions driving the vorticity diffusion and the velocity gradients in the Prandtl boundary-layer.

Let us now consider a set of  $N$  vortex sheet elements  $S_i$  of strengths  $\alpha_i$ . These elements should be seen as segments of a straight line, parallel to the  $x$  axis and of length  $\varepsilon$ , such that the two values of  $u$  at the extremities of  $S_i$  differ from each other by  $\alpha_i$ . According to this definition, the vortex sheet approximation of  $\omega$  is given by:

$$\omega \simeq \sum_j \alpha_j b_\varepsilon(x - x_j) \delta(y - y_j), \tag{68}$$

where  $b_\varepsilon$  is a one-dimensional cutoff function given by  $b_\varepsilon = \varepsilon^{-1} b(x/\varepsilon)$  and where  $\delta$  is the one-dimensional Dirac mass in the normal direction.

The  $u$  component of  $S_i$ , due to the presence of the other surrounding segments, is then obtained from a direct integration of Equation (65), using the right-hand side of the above  $\omega$  approximation (Equation (68)):

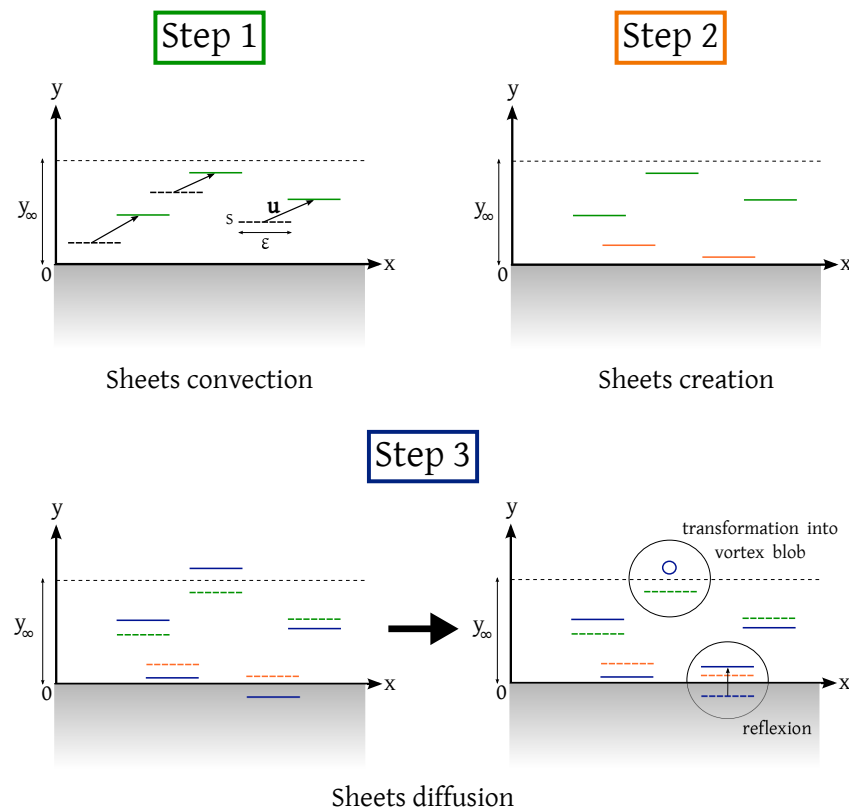
$$u(x, y) = u_\infty(x) + \sum_j \alpha_j b_\varepsilon(x - x_j) H(y_j - y) \tag{69}$$

where  $H$  is the Heavyside function. The expression of  $v$  can then be deduced from  $u$ , using the continuity equation:

$$v(x, y) = -\frac{\partial u_\infty(x)}{\partial x} y - \sum_j \alpha_j b'_\varepsilon(x - x_j) \min(y, y_j) \tag{70}$$

Based on these definitions, one can give the substeps constituting the algorithm of the Chorin’s vortex sheet/vortex blob method (each of these steps are depicted in Figure 7):

1. Evaluation of the  $u$  and  $v$  components from formulations (69), (70) and advection of the vortex sheets with this velocity field.
2. Computation of  $u$  on the wall ( $y = 0$ ) and creation of new sheets at the boundary with strengths  $\alpha_i = -u(x_i)dl$  in order to impose the no-slip boundary condition (these strength values directly come from the integration of Equation (65)). As explained by Chorin in [106], the velocity field is here extended across the wall by the anti-symmetry  $u(x, -y) = -u(x, y)$ . As a consequence  $\omega(x, -y) = \omega(x, y)$  and a vortex sheet is generated at  $y = 0$ .
3. Diffusion of the vortex sheets in the  $y$  direction using a Random Walk approach (see Section 3.1). If a sheet arises in the far-field region ( $y \geq y_\infty$ ), it is transformed into a vortex blob. If a sheet crosses the boundary, moving into the body, it is reflected on the other side.



**Figure 7.** Steps of the vortex sheet/vortex blob algorithm.

In the pioneer version of Chorin’s method [37], the cancellation of velocity at the boundary was realized by generating vortex blobs instead of vortex sheets. But, the overlap of vortex blobs with the solid boundary led to inconsistency. Thanks to its more natural framework implying vortex sheets in the boundary region, this random vortex sheet/vortex blob method turned out to be widely used to model slightly viscous flow past obstacles. For example, one can refer again to the work of Smith and Stansby [107] who studied flows around cylinder, to Sethian and Ghoniem [108] who performed a validation study of this technique for flow past a backward-facing step at different regimes or to Cheer [109] who carried out simulations of flows past airfoils and blunt bodies. One can moreover cite the numerical convergence study of the vortex sheet/vortex blob method carried out by Mortazavi et al. [110] which investigates the influence of three discretization parameters (the blob strength  $\Gamma$ , the core radius  $h$  and the time step  $\Delta t$ ) on the accuracy. The role of the time step has specifically been outlined, with in particular the divergence of the numerical solution for too small values of  $\Delta t$ , due to the Random Walk Method.

The three dimensional version of the vortex sheet/vortex blob method was introduced shortly after, in 1980, by Chorin [111]. In this version, the vortex sheets within the boundary-layer region are now defined as “tiles”. These tiles are rectangles, parallel to the wall and their motion is dictated by the straightforward extension of Equation (65) in 3D. Like in two dimensions, the tiles undergo random walk. They are transformed into vortex blobs (also referred to as vortex segments in 3D, as opposed to vortex filaments) when they leave the boundary layer. This method was successfully used in several studies, for instance by Fishelov [112] or Gharakhani and Ghoniem [113], who respectively dealt with flow past infinite plate and flow inside a cylinder equipped with an eccentric inlet port and a driven piston.

However, the Chorin’s approach suffers from several limits. The first one, already pointed out in the two-dimensional case by Anderson and Reider [114], refers to the interface zone (in which vortex sheet turn into vortex blob) where the continuity of the tangential velocity value is guaranteed but not the one of the normal velocity, which can lead to spurious oscillations close to the interface. Secondly, the divergence free condition

for the vorticity field is only satisfied approximately, which prevents from getting accurate application of vortex methods to three-dimensional flows. Finally, the last and main limitation stands on the fact that the Prandtl approximation makes such approach only valid for attached flows.

As a variant of Chorin’s vortex sheet/vortex blob method, one can cite the approach proposed by Walther and Larsen in 1997 [115] where the determination of the surface vorticity in step 2 is no more based on the Prandtl approximation but on the kinematic relation between the velocity and the vorticity. This method has been successfully applied in 2D to flow past a fixed and harmonically oscillating flat plate at high Reynolds number ( $Re_c = 10^4$ , with  $c$  the chord length of the plate).

### 5.2. Boundary Treatment Based on Vorticity Flux Conditions

The type of method presented in this section was designed later and re-use the framework of algorithms based on viscous splitting and vorticity creation. However, contrary to the Chorin’s method, detailed previously, it handles vorticity flux boundary conditions to enforce the no-slip condition on the walls, instead of vortex sheets or vortex blobs. In a first part we will give the expression of the corresponding continuous problem and secondly we will focus on the numerical schemes and techniques used to approximate the integral solution of this continuous problem.

#### 5.2.1. The Continuous Problem

The idea of a vorticity creation method based on a vorticity flux boundary condition was independently proposed in 1994 by Cottet [116] and Koumoutsakos et al. [117]. The so-called vorticity flux boundary condition is characterized by a relation between the normal derivative of the vorticity and the time derivative of the tangential component of the velocity. To derive the global Navier-Stokes problem, we follow the development described in [116] starting with the 2D Stokes problem. Let us consider two vorticity fields  $\omega_1$  and  $\omega_2$  in the same time interval  $[t^n, t^{n+1}]$ . Each time step is initialized by the exact initial vorticity field  $\omega^n$  and provides at the end the solution  $\omega^{n+1}$  where  $\omega_1$  and  $\omega_2$  are the solutions of the 2 following sub-problems within the interval  $[t^n, t^{n+1}]$ :

$$\text{Sub-problem 1 : } \begin{cases} \frac{\partial \omega_1}{\partial t} - \nu \Delta \omega_1 = 0 & \text{in } \Omega, \\ \omega_1(\cdot, t^n) = \omega^n & \text{in } \Omega, \\ \frac{\partial \omega_1}{\partial \mathbf{n}} = 0 & \text{on } S \end{cases} \quad (71)$$

$$\text{Sub-problem 2 : } \begin{cases} \frac{\partial \omega_2}{\partial t} - \nu \Delta \omega_2 = 0 & \text{in } \Omega, \\ \omega_2(\cdot, t^n) = \omega^n & \text{in } \Omega, \\ \nu \frac{\partial \omega_2}{\partial \mathbf{n}} = -\frac{\partial}{\partial t}(\mathbf{u}_1 \cdot \mathbf{s}) & \text{on } S, \end{cases} \quad (72)$$

where  $\mathbf{s}$  denotes the tangent vector to the surface  $S$  and  $\mathbf{u}_1$  corresponds to the velocity field associated with  $\omega_1$ . In other words, in sub-problem 1, the Stokes equations are solved with an homogeneous Neumann boundary condition, that is to say without creating any vortex sheet on the walls. Then, in sub-problem 2, vorticity is produced at the boundary thus enabling to cancel the slip resulting from sub-problem 1. At the end of the time step,  $\omega^{n+1}$  is set to  $\omega_2(\cdot, t^{n+1})$ . We refer to [116] for the proof of equivalence with Chorin’s algorithm and the proof of convergence using energy estimates. The generalization of the present algorithm to the Navier-Stokes problem is obtained straightforwardly by adding the advection term in Equations (71) and (72) and in replacing the vorticity flux boundary condition of (72) by:

$$\nu \frac{\partial \omega_2}{\partial \mathbf{n}} = - \left[ \frac{\partial \mathbf{u}_1}{\partial t} + (\mathbf{u}_1 \cdot \nabla) \mathbf{u}_1 \right] \cdot \mathbf{s} \quad (73)$$

### 5.2.2. The Integral Formulation

The issue now lies on the resolution of the continuous problems derived above in order to obtain an explicit expression of the approximated vorticity field such that the no-slip boundary condition is satisfied. The general existing algorithms are based on a viscous splitting of the Stokes or Navier-Stokes equations and may be described by the following steps:

1. The Lagrangian advection is performed through the resolution of system (27) and (28),
2. The diffusion is solved in the domain with a PSE scheme for instance (the diffusion scheme needs to be based on a change of vorticity strength),
3. The vorticity flux boundary condition is expressed in terms of an integral formulation derived from the continuous problem (cf. (72) or (73)),
4. The PSE scheme is complemented by distributing this vorticity flux onto the existing vortex blobs so that vorticity enters the domain.

In the last step, particles just experience a change in their strength, they do not multiply themselves like in Chorin’s vortex sheet method. In the following, we focus on steps 3 and 4.

Concerning step 3, the method designed to solve the continuous problem (72) was introduced and validated by Koumoutsakos et al. in [117] and [54]. It is denominated as boundary element method (BEM) or panels method. To present it, one re-writes the problem (72) by considering the vorticity diffusion equation supplemented with an homogeneous initial condition and a Neumann boundary condition:

$$\frac{\partial \omega}{\partial t} - \nu \Delta \omega = 0 \quad \text{in } \Omega \times [0, t], \tag{74}$$

$$\omega(\mathbf{x}, 0) = 0 \quad \text{in } \Omega, \tag{75}$$

$$\nu \frac{\partial \omega}{\partial \mathbf{n}} = F(\mathbf{x}, t) = - \frac{\partial}{\partial t} (\mathbf{u} \cdot \mathbf{s}) \quad \text{on } S \times [0, t]. \tag{76}$$

The integral solution of Equation (76) for the vorticity involves integrals over the surface  $S$  only. To discretize this integral, a boundary integral method is employed by defining  $S$  as a set of discrete panels, denoted with index  $i$ , length  $d$  and centered in  $\mathbf{x}_i$ . Then, the vorticity induced by a panel  $i$  can be given by the following integral:

$$\omega_i(\mathbf{x}, t) = \frac{1}{2} \int_0^t \mu_i(\tau) \phi(\mathbf{x} - \mathbf{x}_i, t - \tau) d\tau, \tag{77}$$

where, if one defines a coordinate system  $\mathbf{x} = (x, y)$  with the  $x$ -axis parallel to the surface, the  $\phi$  function yields:

$$\phi(\mathbf{x}, t - \tau) = \frac{e^{-\frac{y^2}{4\nu(t-\tau)}}}{\sqrt{4\pi\nu(t-\tau)}} \left( \operatorname{erf} \left[ \frac{d+x}{\sqrt{4\nu(t-\tau)}} \right] + \operatorname{erf} \left[ \frac{d-x}{\sqrt{4\nu(t-\tau)}} \right] \right), \tag{78}$$

and where  $\mu(\mathbf{x}, t)$ , the diffusion potential, can be approximated for points  $s$  along  $S$  by:

$$\mu_i(s) \approx -2F_i(s) \left[ 1 - \kappa(s) \sqrt{\pi\nu\Delta t} \right]^{-1} \approx -2 \left( - \frac{(\mathbf{u} \cdot \mathbf{s})}{\Delta t} (s_i) \right) \left[ 1 - \kappa(s) \sqrt{\pi\nu\Delta t} \right]^{-1}, \tag{79}$$

with  $\kappa(s)$  denoting the surface curvature at point  $s$  and  $\gamma_i = - \frac{(\mathbf{u} \cdot \mathbf{s})}{\Delta t} (s_i)$  corresponding to the vortex sheet strength derived from (76). We refer the reader to [117] for a full description of the integral evaluation of Equation (77).

Then one jumps to the step 4 of the algorithm, where the strength  $\Gamma_j = \omega(\mathbf{x}_j)v_j$  of the particles are modified by distributing the vorticity flux:

$$\Gamma_j(t) = \Gamma_j(0) + \frac{v_j \Delta t}{2} \sum_{i=1}^N \mu_i \phi((\mathbf{x}_j - \mathbf{x}_i), \Delta t), \tag{80}$$

with  $N$  denoting the number of panels on the surface of the body. Finally, replacing  $\mu_i$  by approximation (79) in the above equation, the prescription of the no-slip boundary condition is realized through the following particle strength update:

$$\Gamma_j^{n+1} = \Gamma_j^n + v_j \Delta t \sum_{i=1}^N \frac{\gamma_i}{(1 - \kappa(s) \sqrt{\pi v \Delta t})} \phi((\mathbf{x}_j - \mathbf{x}_i), \Delta t). \tag{81}$$

Concerning the 3D formulation, one distinguishes the flat surfaces (half-plane case) from the curved ones.

The first three-dimensional extension of the continuous formulation was given in the case of a flat plate. We note that in 3D, one must ensure that the vorticity is divergence free in order to satisfy the fundamental laws of the fluid mechanics. The direct 3D extension of problems (71) and (72) do guarantee this condition and we refer to Cottet and Koumoutsakos' book [22] for the proof. In order to solve the three-dimensional continuous problems, Casciola et al. [118] proposed a similar analysis as the one described in the above algorithm. First of all, this study supplies the expression of the integral equation for the vorticity creation. An approximation of such integral equation is then given and identified as the slip velocity at the surface of the flat body. Moreover, this work exposes an error estimate of the proposed approximation and also proves that the vorticity generated at the wall and then diffused in the flow is solenoidal.

The expression of three-dimensional continuous problems in the case of non-zero curvatures along the surface  $S$  has been first expressed for cylinders, where the curvature is constant and defined at  $\kappa = 1/R$ , where  $R$  is the cylinder radius. The natural coordinate system to consider herein is the cylindrical coordinate system where one denotes by  $(\mathbf{e}_r, \mathbf{e}_z, \mathbf{e}_\theta)$  the basis vectors, respectively corresponding to the unit normal vector  $(\mathbf{e}_r)$  and the tangential vectors to the cylinder surface  $(\mathbf{e}_z, \mathbf{e}_\theta)$ . According to Poncet and Cottet [119,120] it appears that the divergence free condition is ensured if the Neumann boundary condition  $\partial\omega_\theta/\partial\mathbf{n}$  on the tangential component  $\omega_\theta$  is modified into a Robin type boundary condition, that is to say by  $\kappa\omega_\theta + \partial\omega_\theta/\partial\mathbf{n}$ . The three-dimensional numerical scheme giving the solution of such formulation is derived in the same two references and the validations carried-out herein show the accuracy of the vorticity flux boundary conditions.

In conclusion, the handling of the no-slip boundary condition using vorticity flux formula stands as a suitable and successful method. It indeed allows us to treat in an accurate way the high velocity gradients at the solid-fluid interface while overcoming the main drawbacks of the prior approaches. It is important to highlight the fact that the BEM technique is generally related to the presence of a lattice, first because of the use of a PSE scheme (that needs particle overlapping to ensure convergence, as evoked in Section 3.2) and most of all, because its design is based on the definition of panels on which the flux integrals are discretized. BEM is therefore very often used within Vortex-In-Cell/Vortex-Particle-Mesh algorithms, but not systematically as illustrates the completely mesh-free proposition of Cooper and Barba [121] based on the use of the radial basis function (RBF) collocation method (see Section 4.1). The latter method evaluates the  $\phi$  function of Koumoutsakos (Equation (78)) and solves the diffusion equation by using the radial basis functions and therefore proposes a treatment of the boundary conditions without any panel.

### 5.2.3. Improvements of Vorticity Flux Boundary Conditions

Based on the pioneer works of Koumoutsakos and Leonard presented above, Ploumhans and Winkelmanns proposed substantial improvements in [122] (2D study) and in [55] (3D extension) in the early 2000s. These advances may be separated into two categories.

First, they proposed to improve the step 2 described previously in the viscous splitting algorithm, concerning the diffusion process. Indeed, the authors point out that if the standard PSE scheme is used, namely:

$$\frac{d\omega_j}{dt} = \nu \varepsilon^{-2} \sum_i (v_i \omega_i - v_i \omega_j) \eta_\varepsilon(\mathbf{x}_i^h - \mathbf{x}_j^h), \quad (82)$$

then the particles  $j$  are completely and radially surrounded by other particles  $i$ , which is not true close to the wall. Thus, the use of standard PSE scheme induces spurious vorticity flux at the boundary (even if the total circulation is conserved). To overcome this problem, Ploumhans and Winkelmanns [122] introduced a modified PSE scheme in which *image particles* are defined inside the body when the scheme is applied close to the wall. This process allows a zero vorticity flux during the PSE step, before computing it properly at step 3.

The second type of improvements are based on the fact that BEM (or panels methods) rely on grids to perform the particles remeshing. In Koumoutsakos and Leonard [54], the BEM was proposed in the particular case of a circular cylinder and the particle redistribution was performed on a body-fitted lattice, that followed the cylinder boundary, making it quite restrictive in terms of applications. In their work, Ploumhans and Winkelmanns [122] propose on one side a regular lattice able to redistribute the particles in the case of any general geometry for the immersed obstacle, and on the other side, they introduce a mapping on non-uniform lattices, thus allowing a coarser and controllable resolution for the far wake regions.

All these advances have been validated in the 2D case [122] at different flow regimes for flow past a cylinder, a square and a generic geometry (2D “Apollo” capsule) as well as in the 3D case [55] with the simulation of flow past a sphere at transitional Reynolds numbers (from  $Re_D = 300$  up to  $Re_D = 1000$ ).

### 5.3. Boundary Treatment Based on Immersed Boundary Methods

Driven by the necessity to overcome the difficulties related to the body geometry, the Vortex-Particle-Mesh methods turned, in the early 2000s, to another strategy able to enforce boundary conditions without being constrained neither by the type of underlying mesh nor the obstacle geometry. This strategy lies in their coupling with Immersed Boundary Methods (IBM).

One of the first types of Immersed Boundary Vortex methods to be introduced was an approach in which the boundary conditions are in the form of local forcing terms in the right-hand side of the momentum equations. In this approach, the flow equations are discretized in a unique way in the computational domain, independently of the nature of the cells (i.e., fluid, solid or mixed cells). One can particularly cite the approach proposed in 2003 by Cottet and Poncet [120] in the context of a VIC method, where the no-slip boundary condition is enforced from source points that are located on the boundary itself on all grid points that are at a distance less than the grid-size from the boundary. Few years later, Poncet [123] proposed a revisited immersed boundary technique allowing to satisfy the no-through-flow boundary condition on a body of arbitrary geometry, in an implicit formulation. More precisely, the approach consists in solving the linear system associated to the source-to-flow-through linear application; this system is of reasonable size since it is only constituted by the discretization points of the surface, not all the grid cells. In Poncet’s paper, this vortex IBM is applied to flow past a complete aircraft geometry and to the vorticity evolution around the complex geometry of a hollow bridge structure.



From this first approach arose other immersed boundary vortex methods, presented hereafter.

### 5.3.1. The Brinkman Vortex Penalization Method

The Brinkman penalization technique is part of the continuous forcing immersed boundary methods. It was first proposed in a context completely detached from vortex methods by Caltagirone in 1994 [124], further developed by Angot et al. [125] and was expressed and used in a vorticity formulation by Kevlahan and Ghidaglia in 2001 [126]. The resulting Brinkman-Navier-Stokes vorticity equations are:

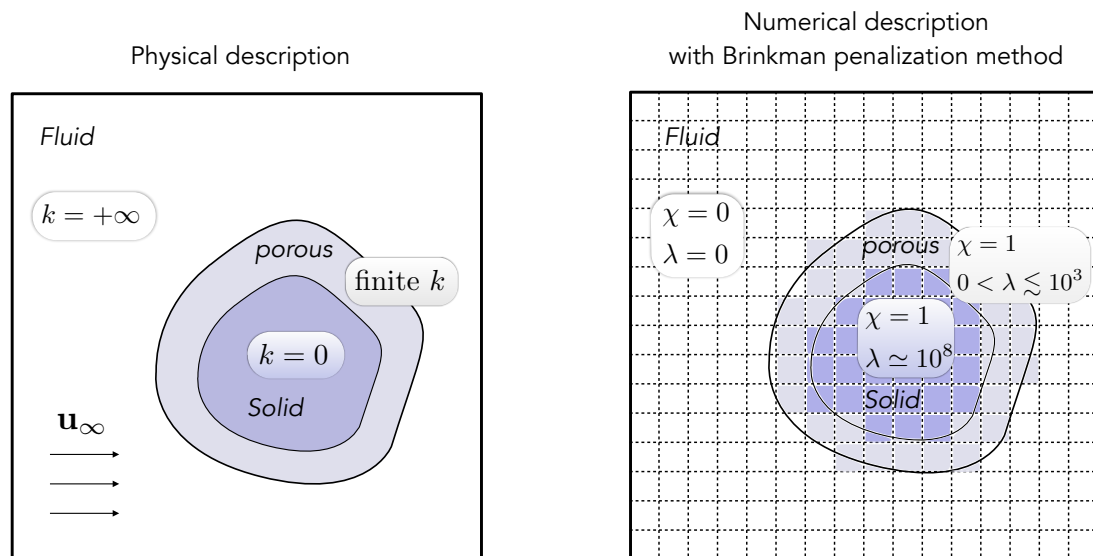
$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla) \omega - (\omega \cdot \nabla) \mathbf{u} - \nu \Delta \omega = \nabla \times [\lambda \chi(\mathbf{u}_s - \mathbf{u})], \quad \text{in } \Omega \times [0, t] \quad (83)$$

$$\operatorname{div} \mathbf{u} = 0, \quad \text{in } \Omega \times [0, t] \quad (84)$$

where the forcing term  $\nabla \times [\lambda \chi(\mathbf{u}_s - \mathbf{u})]$  is the penalization term, in which  $\chi$  denotes the characteristic function of the solid bodies,  $\mathbf{u}_s$  corresponds to the velocity of the immersed obstacle and  $\lambda$  is the non-dimensionalized penalization factor. This penalization term actually comes from the Darcy equations that govern flows in porous media, and the factor  $\lambda$  is directly related, in the inverse proportion, to the permeability  $k$  of the considered media in the domain, i.e.,  $\lambda \sim 1/k$ . The penalization factor  $\lambda$  is set to zero in the fluid region and to infinity in the solid ones. With such approach, the force is activated only inside the solids, making the velocity vanishing in these regions, and the no-slip boundary conditions are automatically satisfied. The condition  $\mathbf{u} = \mathbf{u}_s$  is thus prescribed on the boundary and rigid motion is ensured within the solid.

In the 2010s arose the first resolutions of the vorticity-penalization Equations (83) and (84) through VPM methods. In practice, the penalization equation  $\partial \omega / \partial t = \nabla \times [\lambda \chi(\mathbf{u}_s - \mathbf{u})]$  is added to the fractional step algorithm and is solved on the grid, often using finite-differences and an implicit Euler scheme. In the last decade, this technique has been widely applied within VPMs to handle different types of problems involving boundaries. We can for instance cite the works of Gallizio [96] and Morency et al. [127] who coupled the penalization technique with a level-set method in order to handle moving bodies. Besides, in terms of fluid-structure interactions we can cite the works of Coquerelle and Cottet [128] who also coupled the penalization method with a level-set method and moreover with a collision model to simulate the interaction of an incompressible flow with rigid bodies. One also has to mention all the studies of Gazzola et al. focusing on obstacles' motion induced by vortical flow-structures [10] or dealing with fish schooling [6–8,86], where the penalization method is combined to a projection approach allowing to model the action of the fluid on the deforming fishes (more details are reported in Section 7 about these applications).

In terms of numerical advantages, the vortex penalization method naturally exhibits the ones of the immersed boundary methods, that is to say the absence of explicit prescription of the boundary conditions, the existence of a unique equation for the whole domain  $\Omega$  and the possibility to handle complex geometries. Moreover, the Brinkman penalization method is particularly easy to implement and to parallelize. Finally, it allows the direct modeling of porous medium when giving intermediate values to the penalization factor,  $0 < \lambda \lesssim 10^3$ . Figure 8 depicts and summarizes the modeling of flow in solid-fluid-porous media with the Brinkman penalization method and we refer the reader to Section 7 for an example of related application. However, the major drawback of this approach lies in its first-order accuracy. Besides, performing relevant simulations imposes a restrictive overall time step  $\Delta t$  in the fractional step algorithm.



**Figure 8.** Schematic representation of the physical description (left) and its numerical parametrization (right) for flow past a solid-porous body using the Brinkman penalization method.

To remedy these weaknesses, Hejlesen et al. [129] introduced an iterative penalization scheme allowing to accurately enforce no-slip boundary conditions through sub-iterations of the penalty method. The proposed scheme imposes the resolution of a Poisson equation at each iteration to re-evaluate the penalized velocity, but it is applied only in the vicinity of the body, which on one hand reduces the computational cost of each Poisson sub-problem and on the other hand allows us to consider larger time-steps within the iterative penalization loop than the ones usually required in a standard Brinkman penalization. This iterative approach has proved to be more accurate for all the classical two-dimensional benchmarks tested by the authors (flow past an impulsively started cylinder at  $Re_D = 9500$  and flow past a flat plate at different regimes). Another version of iterative penalization method was also proposed by Gillis et al. in [130] where they formulate the penalization problem as a linear system that is solved iteratively using a recycled Krylov subspaces.

### 5.3.2. The Immersed Interface Vortex Method

Despite the improvements brought by the iterative Brinkman penalization methods described previously, the latter keep on smearing the interface of the immersed body and are still first order convergence methods. To overcome these weaknesses, a third type of immersed boundary technique was proposed in 2016 by Marichal et al. [131] and in 2019 by Gillis et al. [132] to be coupled to VPM: the immersed interface method (IIM). In a general point of view, the immersed interface method (originally proposed by Leveque in 1994 [133]) consists in treating the flow discontinuities at the body's surface through finite-difference stencils, where the jumps are precisely taken into account inside the stencils thanks to modified Taylor series. These jump-corrected Taylor series allow us to evaluate the derivatives of the flow fields, which remain valid across the interface. The prescription of the desired boundary conditions is then realized by injecting them directly in the jump-corrected Taylor series discretizing the derivatives of the velocity field at each control point of the body's surface. In Gillis et al. [132], this approach is embedded in a two-dimensional VPM algorithm and tested in the case of flow past a 2D impulsively started cylinder at high Reynolds numbers (up to  $Re_D = 40,000$ ). On one side the results show a second order spatial accuracy, up to the boundary region, and on the other side they reveal to be in good agreement with previous VPM computations based on Brinkman penalization performed by Rossinelli et al. [134], while considering an eight-fold coarser grid size near the wall.

## 6. Algorithmic Issues

All the sections written so far enabled to outline the landscape of the different families of Vortex Method, with their respective characteristics in terms of numerical convergence, accuracy and reliability. In this part we will now focus on the algorithmic and computational techniques developed to efficiently handle them. This point constitutes a crucial issue for any numerical method which aspires to be considered as a usable and challenging one.

### 6.1. Fast Vortex Methods

Regardless of which type of Vortex Methods one considers, it is admitted that the intrinsically most expensive operation in any vortex algorithm relies on the resolution of the Poisson problem. Indeed, for  $N$  particles, the classical discretization of the Poisson Equation (15) through the Biot-Savart Law (17) implies to compute the interactions between each particle and all the others, leading to a  $\mathcal{O}(N^2)$  computational cost. A wide range of studies have been dedicated to the task of reducing this cost. They are mainly split into two categories: the strategies dedicated to the Vortex-Particle (Lagrangian) framework and the one dedicated to the Vortex-Particle-Mesh context.

Among the algorithms able to solve the Biot-Savart law, the most efficient one is certainly the Fast Multipole Method (FMM). This technique, introduced by Greengard and Rokhlin in 1987 [135], exploits in the context of vortex methods the geometrical distributions of the vortices to evaluate the velocity. The key point is that the velocity field induced by a group of particles clustered around a center does not need to be computed directly from its individual members, but it can be approximated by a finite number of multipole expansions. This approach, based on tree data structures, thus allows us to achieve a  $\mathcal{O}(N)$  computational cost. Parallel multicores FMM implementations have been widely developed and used in the context of Lagrangian Vortex methods, like in the works of Marzouk and Ghoniem [136], Yokota et al. [137] or Giannopoulou et al. [138], to deal with the velocity resolution. Besides its capability to accelerate the evaluation of particle interactions, FMM was also used by Hu et al. [139] to reduce the time consumption of the stretching term calculation in the vorticity equation. However, the use of FMM algorithms is not restricted to Lagrangian Vortex approaches and was also implemented within Vortex-Particle-Mesh methods, as it will be illustrated in the last Section 7.

As one evokes the Vortex-Particle-Mesh methods (VIC, semi-Lagrangian), they particularly and naturally focused on fast grid-based Poisson solvers in order to take advantage of the presence of an underlying mesh. The latter are called fast Poisson solvers, as opposed to the traditional “slow” solvers based for instance on Jacobi, Gauss-Seidel, Conjugate Gradients or Successive Over Relaxation (SOR) methods. These fast Poisson solvers can be defined as hierarchical methods and among them, one can distinguish the direct solvers from the iterative ones. The most popular direct solvers are the Cyclic Reduction method, consisting in discretizing the Poisson equation by means of finite-differences method over a rectangle by a successive splitting of the initial problem, and the Fast Fourier Transform method, which consists in solving the Poisson equation in the Fourier space using FFTs. These two direct grid-based Poisson solvers present a  $\mathcal{O}(N \log N)$  cost and have been widely used in VPM, for instance by Cottet and Poncet [120] for the Cyclic Reduction method (based on the Fishpack solver), or by Mimeau et al. [100] for the FFT approach (based on the FFTW fortran solver), where the computational cost of the Poisson problem was reduced to 5% within a whole time-step. On the other hand, one of the most famous iterative solvers relies on the Multigrid method, which recursively uses coarse discretizations of the Poisson problem to approximate its solution, following V-cycles. The complexity of Multigrid algorithms is  $\mathcal{O}(N)$ , which makes them very attractive in terms of computational cost. However, they may require more efforts in terms of implementation and use.

### 6.2. Performance-Computing Improvements Using GPUs

Because of their close link with computer graphics, the smooth particle hydrodynamics (SPH) methods were the first particle methods to have been implemented on graphic

processors (GPUs) in the mid-2000s. Few years later, in 2008, arose the first implementation of Vortex methods in GPUs with the work of Rossinelli and Koumoutsakos [70]. In this pioneer contribution, the implemented 2D solver relies on a remeshed vortex method and runs exclusively on the GPU, including a periodic 2D multigrid Poisson solver developed for the GPU. This work, validated on 2D vortical flows was further successfully tested on 2D bluff body flows with vortex penalization techniques and a CUDA FFT library (cuFFT) by Rossinelli et al. [140]. It shown a good agreement and, above all, important speed-up with respect to identical simulations performed with prior solver based on CPU (Central Processing Unit) (up to 30 times speed-up of CPU-computations for flows past an impulsively started cylinder). In terms of GPU-simulations based on remeshed vortex methods, one can also cite the very recent work of Keck [141] who managed to solve a three-dimensional sediment flow on a  $1537 \times 512 \times 512$  Cartesian grid with a unique 32 gigabits Nvidia® Tesla V100 node, within only 6 h (this work will be presented into more details in Section 7).

Regarding mesh-free approaches, Yokota and co-authors also developed in the late 2000s a FMM-GPU algorithm (inspired from the one first proposed by Gumerov and Duraiswami in 2008) to realize Lagrangian vortex method simulations of 3D isotropic turbulence [137]. Few years later, they proposed in [142] a comparison between a spectral method and a FMM-based vortex method on 4096 GPUs for the simulation of 3D homogeneous-isotropic fluid turbulence with 69 billions of grid nodes or particles. The authors highlight the limits of spectral methods in simulating problems on such a great number of GPUs and show that, in such configuration, the parallel FFM-GPU solver allows us to achieve a much better computational performance and efficiency than spectral methods, due to the non-local FFT-based operations of the latter, and show the relevance of advanced Lagrangian vortex solvers for the simulation of isotropic turbulence.

### 6.3. Adapted Grids, Multiresolution Aspects and LES Models

The Lagrangian framework of Vortex-Particle methods intrinsically allows us to spatially adapt the particle resolution according to the flow physics (provided the overlapping condition is satisfied). One can refer to the recent work of Giannopoulou et al. [138], based on the DVH method (already evoked in Section 4.1), where a self-adaptive particle resolution (also called *r*-adaptivity) is carried out in order to progressively reduce the particle size when getting closer to the boundaries. The method is tested in the cases of two-dimensional flows past cylinders with circular, elliptical and triangular sections, and successively compared to a Finite-Volume method. However the results reveal a higher computational cost with the DVM approach, which can be in particular explained by the large number of vortices required at the boundary compared to the number of cells required in Finite-Volume methods as well as the smaller time steps required by the explicit DVH approach.

Concerning hybrid vortex methods (i.e., Eulerian/Lagrangian domain decomposition or VPM methods like VIC and semi-Lagrangian vortex methods), they have encountered, as shown throughout this paper, a growing notoriety since the late 1990s and rapidly got concerned by the cost of the grid involved in their schemes. Like in Eulerian methods, questions related to adapted grids, multiresolution simulations or subgrid scale models therefore naturally arose in order to give VPM methods the opportunity to handle complex physical problems and/or turbulent flows, which require a high level of mesh refinement and thus a high computational cost.

#### 6.3.1. Adaptive Mesh Refinement (AMR)

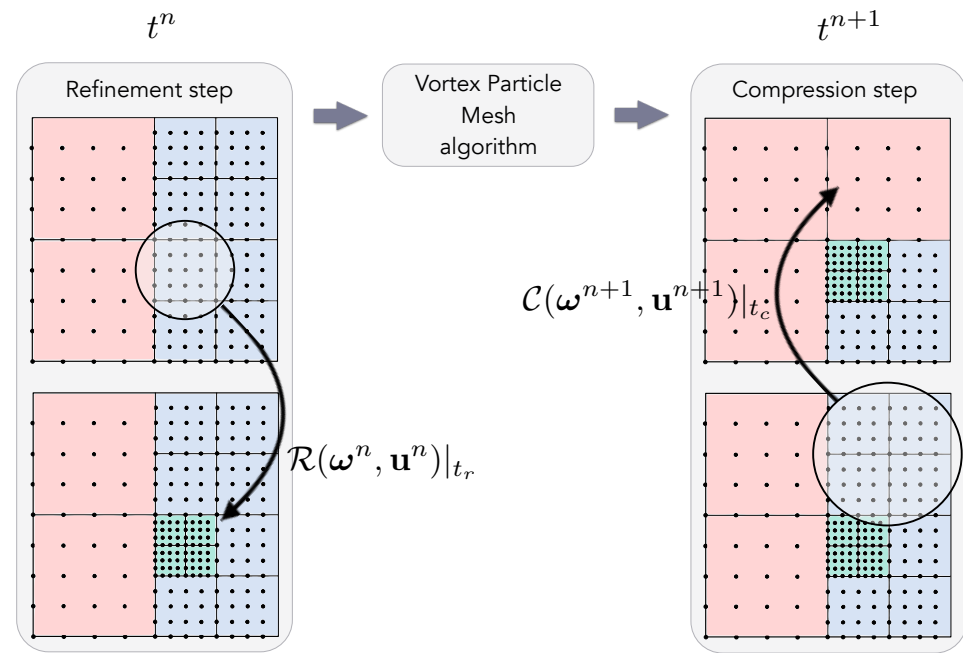
This technique consists in defining blocks of uniform grid sizes in a dynamic way from *a posteriori* estimates. It has been first introduced within a 2D remeshed vortex method by Bergdorf et al. in 2005 [143]. In this work the two-dimensional grid adaptivity obeys a criterion based on the velocity derivatives in order to refine the underlying mesh in the flow regions of high gradients. The simulation results are compared to those of a *r*-adaptive

vortex method with global mappings for the inviscid evolution of an elliptical vortex of compact support in an unbounded domain. It turns out that AMR-based method seems to be usable in a wider range of problems since the refined regions can be created on demand based on any criterion (provided a stepwise refinement transition from coarse to fine grid-sizes), contrary to the method based on adaptive global mappings that must adapt smoothly to the flow physics involved in the problem. Rasmussen et al. [144] also proposed a 2D multiresolution VIC algorithm with Brinkman penalization and validated it in the case of classical benchmarks involving bluff bodies; in this study the local refinement is not adaptive but fixed *a priori*. In 2010, El Ossmani and Poncet [145] introduced a 3D adaptive mesh refinement based on multilevel sub-blocks that adapt to the flow evolution through the periodic evaluation of the quantity  $\alpha|\omega| + \beta|\nabla \times \omega|$ . They tested it to study 3D ellipsoid vortex dynamics and the wake development behind a sphere.

Based on these considerations, Bergdorf and Koumoutsakos [87] proposed in 2006 another multiresolution criteria for the grid adaptivity within remeshed vortex methods: the wavelets. The decomposition of a signal into wavelets (i.e., wave-like oscillations) is a mathematical tool complementary to Fourier transform. It is often used to extract information from data and to design compression/decompression algorithms. In the context of multiresolution analysis, the wavelet theory is applied to extract information from velocity and vorticity fields and to induce local refinements or compressions of the grid.

More precisely, the wavelet-based adaptivity proposed in [87] is performed using biorthogonal wavelets. The latter introduce a pair of analysis functions (involving the Fast Wavelet Transform) and a pair of synthesis functions (involving the inverse FWT). The velocity and vorticity field can therefore be decomposed in “scaling” and “detail” coefficients at different levels of resolution. Each point of the adaptive grid is thus represented by a scaling and a detail coefficient. The aim is to discard the detail coefficients which, compared to a fixed threshold, do not carry relevant information about the flow field. For each level of resolution, one only keeps the coefficients called active coefficients. To build up the wavelet-based AMR method, a block structure is used. Each block contains active coefficients situated at the same level of resolution and all blocks contain exactly the same number of active coefficients. These blocks are built so that they can be split into finer (refinement) or collapsed into coarser blocks (compression). In order to capture the emerging small scales, refinement of the grid is performed, written  $\mathcal{R}(\omega^n, \mathbf{u}^n)|_{t_r}$ , based on a refinement threshold  $t_r$  at the beginning of each time step through inverse FWT. For compression, noted  $\mathcal{C}(\omega^{n+1}, \mathbf{u}^{n+1})|_{t_c}$ , a threshold  $t_c$  is used at the end of each time step to truncate terms in reconstruction, retaining active coefficients and discarding detail coefficients which do not carry significant information. The blocks where these detail coefficients are located then collapsed into coarser ones performing FWT. Figure 9 depicts this process. The concept of wavelet adaptivity has been later implemented by Bergdorf, Koumoutsakos and co-authors in a 2D remeshed vortex library, called MRAG-I2D [134], where the space-adapted grids are also coupled with Local Time-Stepping (LTS) integration schemes in order to speedup the computational efficiency. These schemes, exploiting the time step locality imposed by the different stability conditions, allow us to integrate the coarser elements with larger time steps than the finer ones and thus enable one to perform fewer integration steps. Local time-stepping schemes are shown to speedup the computation by one order of magnitude at each level of resolution from a certain number of levels. The non-uniformity of the grid imposes the use of a Fast Multipole Method (FMM) to solve the velocity Poisson equation. The MRAG-I2D library efficiently maps these different algorithmic and numerical aspects onto heterogeneous CPU-GPU platforms. It has been used for the simulation of bluff body flows with Reynolds numbers ranging until 40,000, flow-mediated interactions between two solids, analysis of optimal shapes in anguilliform swimming and reinforcement learning in the context of self-propelled bodies (these applications will be further detailed in Section 7).





**Figure 9.** Sketch of refinement/computations/compression process within one time step of Vortex-Particle-Mesh method with wavelet-based Adaptive Mesh Refinement (AMR).

### 6.3.2. Subgrid Scale Vortex Methods

When the flow under study becomes turbulent, and therefore computationally expensive to simulate, the Large Eddy Simulations (LES) allow us to only capture and resolve the most energetic scales while modeling the effect of the Sub-Grid Scales (SGS) on the captured scales [146]. Before presenting some SGS vortex methods, let us first briefly introduce the general concept of LES methods. As evoked above, this approach consists in only resolving the smallest scales down to the cutoff length  $\Delta$  of a chosen filter  $\mathcal{F}$ . Typically the spatial discretization is on the order of  $\Delta$ . Thus the resolved (or filtered) quantity  $\tilde{Q}$  is defined as:

$$\tilde{Q} = \mathcal{F}_\Delta \star Q(\mathbf{x}, t), \tag{85}$$

and the remaining smallest scales must be modeled. The filtered Navier-Stokes equations, written here in their velocity-pressure formulation, therefore can be written:

$$(\text{NS}_{LES}) : \begin{cases} \frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\tilde{\mathbf{u}} \cdot \nabla) \tilde{\mathbf{u}} = -\frac{1}{\rho} \nabla \tilde{p} + \nu \Delta \tilde{\mathbf{u}} - \nabla \cdot \boldsymbol{\tau} \\ \text{div}(\tilde{\mathbf{u}}) = 0 \end{cases} \tag{86}$$

where  $\tau_{ij} = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j$ , referred to as the subgrid scale stress tensor, models the smallest scales behavior in the flow. Among the different classes of subgrid scale models, the eddy-viscosity models are the most commonly used as for instance the famous Smagorinsky model, given by:

$$\tau_{ij} = -2\nu_{sma} \tilde{S}_{ij} \quad \text{with} \quad \nu_{sma} \equiv (C\Delta)^2 |\tilde{S}| \tag{87}$$

where  $C$  is a fixed constant, called the Smagorinsky coefficient, and  $|\tilde{S}|$  is the rate-of-strain tensor.

Vortex methods allow us to carry out under-resolved viscous simulations and large eddy dynamics. To illustrate this statement we will describe the guidelines of three different approaches.

The first one is the anisotropic subgrid scale model, based on Vortex method, proposed by Cottet in 1996 [147]. The 2D model derivation is based on the resolution of the regularized/smoothed Euler equations using vortex blob method (see Section 2.1):

$$\frac{\partial \omega}{\partial t} + \text{div}(\mathbf{u}_\varepsilon \omega) = 0, \tag{88}$$

$$\mathbf{u}_\varepsilon = K_\varepsilon \star \omega \quad \text{with} \quad K_\varepsilon = K \star \zeta_\varepsilon, \tag{89}$$

where the function  $\zeta_\varepsilon = \frac{\zeta(\mathbf{x}/\varepsilon)}{\varepsilon^2}$  results from a cutoff function  $\zeta$ , that one assumes positive and decaying at infinity, of mean value 1 and with radial symmetry. The approximation of the exact vorticity field  $\omega$  is then defined by the regularization  $\omega_\varepsilon = \omega \star \zeta_\varepsilon$ . In the spirit of LES methods, at a finite core size  $\varepsilon$  it is appropriate to interpret  $\omega_\varepsilon$  as the filtered vorticity, and the core smoothing function  $\zeta_\varepsilon$  as the spatial filter, of size  $\Delta \equiv \varepsilon$ . Performing the convolution of (88) with  $\zeta_\varepsilon$  one gets:

$$\frac{\partial \omega_\varepsilon}{\partial t} + \text{div}(\mathbf{u}_\varepsilon \omega_\varepsilon) = E(\mathbf{x}), \tag{90}$$

where  $E$  represents the truncation error resulting from the fact that the convolution was only applied on  $\omega$  in the advection term  $\text{div}(\mathbf{u}_\varepsilon \omega)$ . Based on the expression of error  $E$ , one can then derive an evaluation of the enstrophy production, given by:

$$\frac{d}{dt} \int \omega^2 d\mathbf{x} = - \iint [\omega(\mathbf{x}) - \omega(\mathbf{y})]^2 [\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y})] \cdot \nabla \zeta(\mathbf{x} - \mathbf{y}) d\mathbf{x}d\mathbf{y}. \tag{91}$$

where the subscript  $\varepsilon$  has been dropped for clarity. This expression indicates that the enstrophy will increase because of the exchange of vorticity between point  $\mathbf{x}$  and  $\mathbf{y}$  when they will satisfy:

$$[\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{y})] \cdot \nabla \zeta(\mathbf{x} - \mathbf{y}) < 0. \tag{92}$$

meaning that the truncation error  $E(\mathbf{x})$  in the mollified Euler equations intrinsically embed antidiffusive (or backscatter) mechanisms. The idea proposed in [147] is therefore to clip antidiffusivity while leaving contributions in the direction of dissipation untouched, leading to the following 2D anisotropic eddy viscosity model:

$$\frac{d\omega_p}{dt} = \varepsilon^{-2} \sum_q v_q (\omega_p - \omega_q) \{ [\mathbf{u}(\mathbf{x}_p) - \mathbf{u}(\mathbf{x}_q)] \cdot \nabla \zeta(\mathbf{x}_p - \mathbf{x}_q) \}_- \tag{93}$$

where the index—denotes the negative part of the quantity between brackets. This diffusion scheme, very reminiscent of PSE methods, is conservative. As explained by Cottet in [147] and in all the following works of the same author related to this subject, the proposed eddy viscosity model turns out to be more accurate than usual Smagorinsky-type models, which would produce stronger diffusion for identical flow problems.

A few times later, Mansfield et al. [148] proposed another development of a LES vortex scheme in order to model the effects of subfilter scale (SFS) velocity and vorticity fluctuations. To do so, the authors start from the 3D filtered vorticity transport equation:

$$\frac{\partial \tilde{\omega}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{\omega}_i}{\partial x_j} = \tilde{\omega}_j \frac{\partial \tilde{u}_i}{\partial x_j} + \nu \Delta \tilde{\omega}_i - \frac{\partial R_{ij}}{\partial x_j} \tag{94}$$

where  $\tilde{(\cdot)}$  denotes the low-pass filtered quantities and  $R_{ij} = (\tilde{\omega}_i \tilde{u}_j - \tilde{\omega}_i \tilde{u}_j) - (\tilde{u}_i \tilde{\omega}_j - \tilde{u}_i \tilde{\omega}_j)$  is the subgrid scale vorticity stress.  $\partial R_{ij} / \partial x_j$  refers to the subfilter scale torque  $\mathbf{g}$ . This SFS torque is modeled based on a Smagorinsky-like eddy diffusivity scheme:

$$\mathbf{g} = -\nabla \times (\nu_{sma} \nabla \times \tilde{\omega}) \tag{95}$$

with  $\nu_{sma} = (C\Delta)^2 |\tilde{\mathcal{S}}|$  the Smagorinsky eddy diffusivity. The authors then propose a particle discretization of this SFS model based on a vortex blob discretization and introduce a dynamic variation of the model coefficient  $C$ , directly evaluated in a Lagrangian way from the particles data. Based on this approach, the same authors performed dynamic LES of colliding three-dimensional vortex rings in [149]. The results coincide with experimental observations and highlight in particular the capability of the model to correctly simulate



the generation of small-scale turbulent structures as well as the formation of small-scale ringlets propagating radially after collision. The same approach was also used by Pinon et al. [150] (with a fixed model-coefficient  $C$ ) in order to perform LES of a 3D flow of a round-jet in a crossflow, providing results in good agreement with previous experimental and numerical studies.

With the increasing use of Vortex-In-Cell and semi-Lagrangian vortex methods, a novel way of performing LES within vortex methods appeared, where the subgrid scale vorticity stress in the rhs of the filtered vorticity transport equation is directly solved on the grid, using finite-differences schemes. This is what proposed Cocle et al. [151] who investigated several types of existing SGS models within a VIC method. Among the tested models, the usual Smagorinsky model is not able to be activated only during the complex phases of the flow, and turns out to be too diffusive, as expected. A good compromise highlighted in [151] is the use of the so called “regularized variational multiscale” model. In this approach, the artificial diffusion does not operate on the complete LES vorticity field, as in usual Smagorinsky models, but acts solely on the high frequency part of the LES vorticity field:

$$\frac{d\omega}{dt} = \nabla \cdot (\mathbf{u}\omega) + \nu\Delta\omega + \nabla \cdot (v_{sma}(\nabla\omega^s + (\nabla\omega^s)^T)) \tag{96}$$

with  $\omega^s$  denoting the "small-scale" vorticity field defined as  $\omega^s = \omega - \tilde{\omega}$  (where  $\tilde{\omega}$  is the filtered vorticity obtained through a discrete filter of size  $\Delta$ ) and where the subgrid viscosity  $v_{sma}$  is evaluated through a Smagorinsky model (see definition (87)). As reported in [151], this multiscale subgrid model included in a VIC method allows us to perform large eddy simulations with good spectral behavior. It indeed preserves the low and medium wavenumbers while ensuring the SGS dissipation at the high frequencies. This SGS model has been further used within others VPM methods, to handle challenging applications in aerodynamics (some of them are presented in Section 7).

All the algorithmic strategies exposed in this section, as well as the cited related works, are summarized in the diagram of Figure 10.

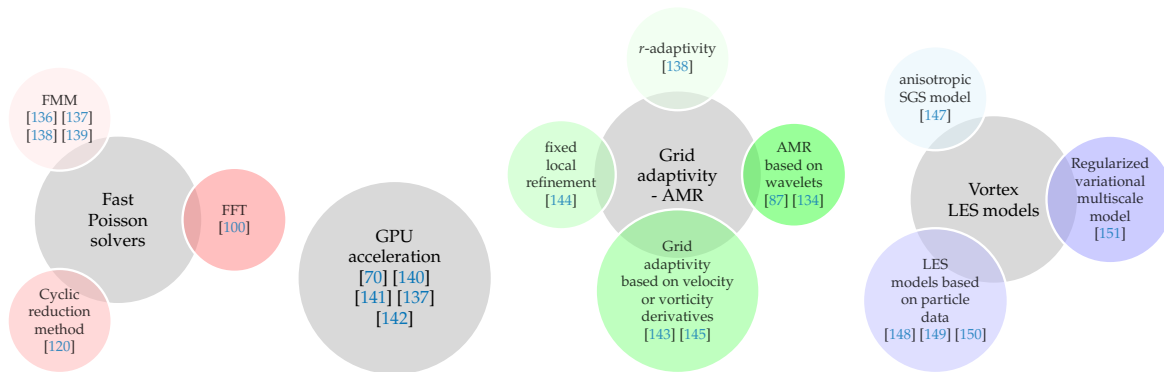


Figure 10. Algorithm issues in Vortex methods: summarize-diagram of Section 6.

### 7. Applications and Issues

Previous section illustrates the capability of purely meshless vortex methods to excel in the calculation of isotropic turbulence. They also proved their ability in simulating simple geometrical bluff body flows. Besides these types of applications, they seem to show a certain difficulty in handling complex flow physics implying the coupling of the Euler or Navier-stokes equations with other models.

On the contrary, hybrid vortex methods like Eulerian/Lagrangian domain decomposition or VPM methods, continually succeeded during the last two decades in undertaking more and more challenging applications. This last section proposes a non-exhaustive presentation of recent and advanced simulations performed with hybrid vortex methods in order to illustrate their capacities and their versatility. Table 1, given at the end of this

section, outlines the main physical and numerical characteristics related to each of them.

### 7.1. Hydrodynamics

The simulations exposed hereafter deal with Vortex-Particle-Mesh methods used to study fluid-structure interactions in hydrodynamics and more specifically in fish schooling. The involved Reynolds numbers (based on the fish length  $L$ ) stand in the range  $100 \leq Re_L \leq 5000$ .

**Propelling and energy harvesting of articulated bodies** (referred with number (#1) in Table 1) by Bernier et al., 2019 [9]. This study examines the hydrodynamics resulting from passive and actuated bodies in fluids. More precisely it consists in handling the passive propulsion of a free swimming articulated fish within the wake of a leading bluff body. The free swimmer is modeled by a multi-body articulated system including complex kinematic constraints like rotational joints and linear rails for self-propulsion as well as damper-like elements for energy harvesting. The parametric simulations show an efficient combination between power production (for the damped passive multi-body swimmer) and flow control (for the leading body).

**Optimization of fish schooling** (#2) by Gazzola et al., 2012b [86] and van Rees et al. 2013 [7]. The simulations performed in these two studies both include a CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy) optimization algorithm within their VPM method in order to study optimal swimming regimes for anguilliformed robots. For the modeling of the swimming bodies, both study involve independent local deformable grids on which are defined the fish shapes and carrying the values of deformation velocities and characteristic/level-set functions, then interpolated on the underlying Cartesian computational grid. In [86], the CMA-ES algorithm is used to identify the fish shape-pattern which maximizes the escape distance. The resulting best motion kinematics exhibit the C-shape pattern as the optimal one, as confirmed by in vivo observations. In [7], the CMA-ES procedure is applied to the identification of the fastest and most efficient shapes for self-propelled swimmers. The results provide achievable robotic-shapes that allow us to outperform real larval zebrafish geometries by 40% and 135% for speed and efficiency, respectively.

**Reinforcement learning for self-propelled swimmers** (#3) by Novati et al., 2017 [152]. Another AI model-free procedure used by Novati et al. within a VPM method to optimize fish swimming lies in reinforcement learning (RL) algorithms. Their simulations involve two fishes, swimming in a synchronised tandem configuration (leader-follower). The coupling of a VPM with a Q-learning (RL) algorithm allows us to automatically define an optimal policy allowing the follower to maximize a desired and specified long-term reward (like the reduction in energy spent).

### 7.2. Aerodynamics

Vortex methods are known to be intrinsically well suited for advection-dominated flows. Problems related to aerodynamics are therefore natural applications for particle approaches. This subsection presents some studies showing the capability of hybrid vortex methods to perform LES of turbulent aerodynamic flow, characterized by Reynolds numbers higher than  $10^4$ .

**Dynamics of the wake of aircraft wings** (#4) by Cocle et al., 2008 [14]. In this study the LES simulations, performed from a VPM method based on the SGS model (96), handle the 3D space-development of wake vortices behind an elliptically loaded wing in ground effect at  $Re_c = 10,000$ . They clearly depict the complex physics occurring in the wake region between primary and secondary vortex and the apparition of small-scales vortices due to the break down of the primary vortex under the effects of the secondary vortex spiral motion.

**Dynamics of rotorcraft wakes** (#5) by Caprace et al., 2020 [12]. Large eddy simulations of a 3D flow past an advancing helicopter rotor are realized through a VPM approach based on SGS model (96) and coupled with an Immersed Lifting Line (ILL) method. The latter, derived from the algorithms commonly used in wind energy community, is fully compatible with the Vortex-Particle-Mesh framework. It indeed consists in computing

the vorticity corresponding to the local forces developed by the modeled wing/rotor and then in introducing this vorticity in the flow using vortex particles, that is to say in a Lagrangian fashion. The simulations performed in this study allowed, in particular, to precisely describe the vortex dynamics involved in the generation of the two main wake vortices of the rotorcraft thanks to thorough 3-D visualizations. For further details about the ILL approach in a VPM framework, one also refers to [99], by the same authors.

**Dynamics of rotorcraft and finite aircraft wakes (#6)** by Stock et al., 2010 [13]. Another type of large eddy simulations of rotorcraft wakes was earlier proposed by Stock et al. using a hybrid Eulerian/Lagrangian decomposition domain approach (see Section 4.3). In the near-body region the computation is realized through a fully-compressible, Eulerian, Navier-Stokes flow solver based on a body-fitted structured grid. In the wake region, a pure Lagrangian vortex method is used, with the following characteristics: the velocity evaluation is performed through a GPU-FMM solver; since the Reynolds numbers considered in this study are in the order of  $Re_c \approx 10^6$ , no diffusion scheme is applied; a boundary element method (BEM) is used to set on each triangular panels the bound vortex sheet strength involved in velocity computation; an anisotropic SGS model based on Cottet's model [147] (3D extension of Equation (93)) is implemented for LES; finally, an overlap region and a coupling algorithm is carried out at the Eulerian/Lagrangian interface. LES of finite aircraft wing wakes are first performed, showing results that coincide rather well with experiments (showing however discrepancies in the region of the tip-vortex core), before presenting LES of 4-blades advancing rotor wakes for two angles of rotation of the whole device in forward flight.

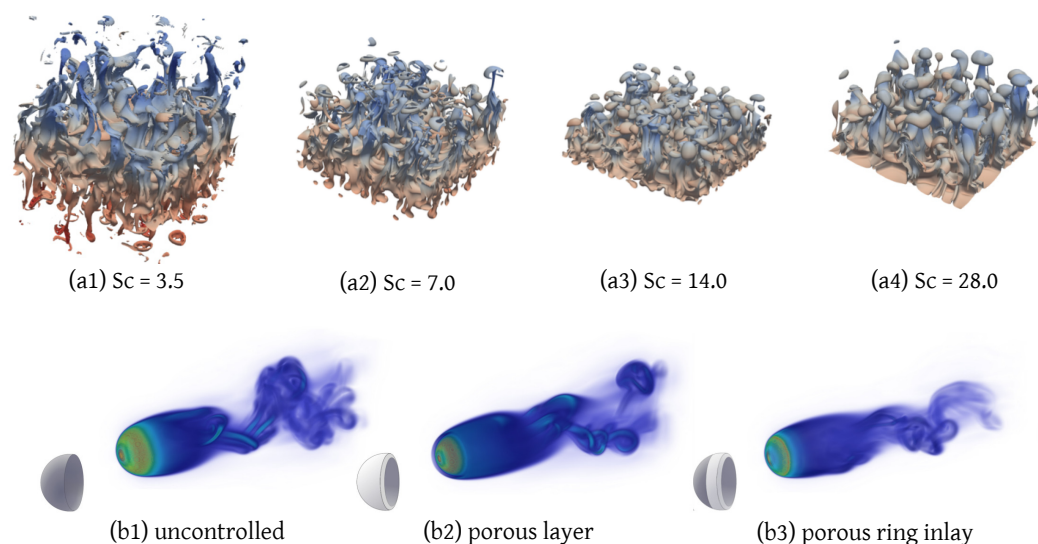
### 7.3. Scalars Transport

Scalar advection problems involve the coupling of the Navier-Stokes equations with an advection-diffusion equation for the scalar transported at the flow velocity. In such problems, the production of small scales is dictated by the value of the Schmidt number,  $Sc = \nu/\kappa$ , defined as the ratio between the viscosity of the fluid and the diffusivity of the scalar. If  $Sc > 1$ , the size of the smallest scalar fluctuations is smaller than the smallest scale of the turbulent flow. In such situation, it appears quite natural therefore to perform numerical simulations involving different grid resolutions for the scalar and the momentum. The two following studies showcase the capability of VPM methods, more precisely here semi-Lagrangian vortex methods (see Section 4.3), to efficiently handle such problems.

**Multi-scale solver for scalar turbulent transport on CPU-GPU (#7)** by Etancelin et al. 2014 [153]. The computing strategy of this study is based on a semi-Lagrangian vortex method and consists in solving the Navier-Stokes equations (momentum and Poisson equation) on multi-CPU while the scalar transport is computed on several GPUs. Passive scalar in a turbulent plane jet of  $Sc = 10$  and  $Re_w = 1000$  (with  $w$  the jet width) is simulated in a hybrid CPU/GPU computation with  $128^3$  and  $512^3$  grid points for the velocity and the scalar, respectively. In [154] a thorough HPC study is performed by the same author for turbulent jet problems (see Figure 6.13, p. 133) varying the respective grid sizes of the flow quantities and the scalar and varying the number of CPUs and GPUs in the parallel computation. It shows in particular that the computational time for the resolution of the scalar on a  $1024^3$  grid on 8 GPUs takes half the time needed for the resolution of the flow on a  $256^3$  grid on 32 CPUs.

**Multi-scale solver for sediment transport on GPU (#8)** by Keck 2019 [141]. This study examines the three-dimensional dynamics of sediment-laden fresh water above salt water. The modeling of such problem involves the coupling of the Navier-Stokes equations with two transport-diffusion equations concerning respectively the salinity  $S$  and the sediment-particles concentration  $C$ . The Schmidt numbers considered in this study for the concentration  $C$  are  $Sc_{(C)} = 3.5, 7, 14, 28$  and the ratio between the two scalars diffusivities is equal to 25. The problem is solved via a semi-Lagrangian vortex method, using GPU based computations. The obtained 3D sediment isocontours show, as expected from literature, salt fingers initially present on both sides of the sediment-laden water/salt

water interface that then disappear under Rayleigh-Taylor instabilities for  $Sc_{(C)} = 3.5$  and 7. For the two other Schmidt numbers  $Sc_{(C)} = 14$  and 28, no more sediment fingers can be seen. They are replaced by downward moving sediment plumes of mushroom-shape, characteristic of Rayleigh-Taylor instabilities (see Figure 11 (a1 to a4)). These computations are performed on a lonely GPU core. An extension of this study to multi-scale hybrid CPU-GPU computations is envisioned.



**Figure 11.** From a1 to a4 (case #8) *Courtesy of Keck [141]* Sediment isocontours showing moving fingers (downward w.r.t interface) at different Schmidt numbers. From b1 to b3 (case #9) Effect of porous-coating configuration on the wake of flow past a hemisphere at  $Re_D = 1000$ .

#### 7.4. Porous Flows

The last examples given in this non-exhaustive list of applications focus on the ability of VPM, and more precisely hereafter of semi-Lagrangian vortex methods, to handle porous flows. This can be done through the coupling of the governing incompressible flow equations with Darcy or Brinkman equations, which are solved on the grid.

**Passive flow control using porous media (#9)** by Mimeau et al., 2014 and 2017 [20,155]. Based on the results obtained in the 2D case [155], the study [20] proposes to simulate 3D incompressible flows in solid-porous-fluid media through a semi-Lagrangian vortex method and then to perform passive control of bluff body flows by partially covering the body surface with a porous coating. The modeling of the physical problem is directly realized through the Brinkman-Navier-Stokes equation (Equation (83)) where the solid, porous and fluid regions are determined by defining characteristic functions and by setting specific values of the penalization factor  $\lambda$  in each region, knowing that intermediate values of  $\lambda$  enable to model a porous medium in which the flow has a Darcy velocity. The method is applied to the passive flow control of a hemisphere at transient Reynolds numbers. The results show that the effects of the added porous layer on the wake stabilization and drag reduction noticeably depend on the permeability and the location/geometry of the covering layer. The parametric simulations allow us to identify an efficient “porous ring-inlay” control device whose porous coating is only located in flow separation areas (see Figure 11 (b1 to b3)).

**Table 1.** The main physical and numerical characteristics of the simulations presented in Section 7.

	Application	Flow Regime	Governing Equations	Numerical Method	Particle Remeshing	Poisson Solver	Boundary Treatment	Grid Type	Simulation	Processor Type
#1 [9]	Swimming of articulated bodies	$Re_L = 100$ , $Re_L = 200$	Navier-Stokes equations	VPM with multi-body solver	Tensorial $M'_4$ scheme	FFT solver	Brinkman penalization	Cartesian uniform	2-D DNS	CPU
#2 [7,86]	Optimization of fish schooling	$Re_L = 500$	Navier-Stokes equations	VPM with CMA-ES algorithm	Tensorial $M'_4$ scheme	FFT solver	Brinkman penalization	Cartesian uniform	3-D DNS	CPU
#3 [152]	Reinforcement learning for swimmers	$Re_L = 5000$	Navier-Stokes equations	VPM with RL algorithm	Tensorial $M'_4$ scheme	FMM solver	Brinkman penalization	Wavelets-based AMR	2-D DNS	CPU
#4 [14]	Dynamics of aircraft wings wake	$Re_c = 10^4$	Navier-Stokes equations	VPM based on SGS model	Tensorial $M'_4$ scheme	FMM solver	-	Cartesian uniform	3-D LES	CPU
#5 [12]	Dynamics of rotorcraft wakes	$Re_c = 4 \times 10^5$	Navier-Stokes equations	VPM based on SGS and ILL model	Tensorial $M'_4$ scheme	FFT solver	ILL method	Cartesian uniform	3-D LES	CPU
#6 [13]	Dynamics of rotorcraft and aircraft wakes	$Re_c = 4.6 \times 10^5$ , $Re_c = 1.5 \times 10^6$	Navier-Stokes equations	Hybrid method based on SGS model	-	FMM-GPU solver	BEM method	Body-fitted grid (near-body)	3-D LES	CPU-GPU
#7 [153]	Passive scalar turbulent transport	$Re_w = 1000$ $Sc = 10$	Navier-Stokes equations	Semi Lagrangian VM	Directional $\Lambda_{6,4}$ scheme	FFT solver	-	Cartesian uniform	3-D DNS	CPU-GPU
#8 [141]	Sediment flow	$Sc = 3.5, 7, 14, 28$	Navier-Stokes equations	Semi Lagrangian VM	Directional $\Lambda_{4,2}$ scheme	FFT solver	-	Cartesian uniform	3-D DNS	CPU-GPU
#9 [20,155]	Flow control using porous media	$Re_D = 300$ , $Re_D = 1000$	Navier-Stokes equations	Semi Lagrangian VM	Directional $\Lambda_{4,2}$ scheme	FFT solver	Brinkman penalization	Cartesian uniform	3-D DNS	CPU
#10 [156]	Rock dissolution at pore-scale	Dissolution regime ( $Pe = 0.7$ )	Darcy Brinkman-Stokes equations	Semi Lagrangian method (for reaction equations)	Directional $M'_4$ and $M_4$ schemes	FFT solver	penalization	Cartesian uniform	3-D DNS	CPU-GPU



**Carbonate rock dissolution at pore-scale (#10)** by Etancelin et al., 2020 [156]. Contrary to all the previous studies reported in this review paper, this last illustration of semi-Lagrangian capabilities deals with Darcy-Brinkman-Stokes equations, that is to say with problems not dominated by advection. More specifically, this last study aims to simulate dissolution processes at the pore scale of actual rocks. To do so, the authors couple a Lagrangian formulation of the chemical reaction equations (handled through a semi-Lagrangian solver) with the so called superficial velocity formalism (handled with an Eulerian scheme). The authors address the crucial issue of how to estimate the numerical diffusion induced by the remeshing step in the semi-Lagrangian solver and how to switch from a low-order to a high-order kernel in order to correctly adapt the physical diffusion. The 3D DNS simulations highlight the effects of fluid inclusions in a non-percolating and percolating carbonate rock.

## 8. Conclusions and Perspectives

### 8.1. Conclusions

The intensive development of the vortex methods between 1970 and the mid 1990s was motivated by the will to design robust and accurate numerical schemes able to overcome the main weaknesses of traditional grid methods, especially in the case of dominant advection effects in the flows. The major advantages arising from these pure Lagrangian methods are the following: appealing physical attributes, automatic adaptivity of the discretization elements (particles), compactness of vorticity support, accurate handling of boundary conditions at infinity, negligible numerical dissipation, conservation of the flow invariants, relaxed stability condition on the time step and usefulness of vorticity in terms of CFD results interpretations. However, during this period, vortex methods experienced some difficulties. First they had to face an inaccurate modeling of the viscous effects due to the particle distortion phenomenon. They were also confronted with the expensive resolution of the Poisson equation allowing to recover the velocity field from the vorticity. Finally, they failed in providing both accurate, efficient and generalized treatment of boundary conditions on solid walls due to the velocity formulation of the no-slip boundary conditions. For all these reasons, and despite the efforts made to design solutions to overcome the difficulties, these solutions remained partial and the purely Lagrangian vortex methods did not manage to become a common tool for the simulation of incompressible real and complex flows. In order to bypass their intrinsic limitations, they started to move towards hybrid formulations in the early 1990s, resulting in the introduction of underlying grids with remeshing. The presence of Cartesian grids indeed enables the use of Fast Poisson solvers as well as grid-based schemes for the discretization of the diffusion term, and makes possible the handling of immersed boundary methods for the treatment of the solid bodies, while ensuring a permanent control of the particle overlapping through regular redistribution operations on the grid. In the meantime, the particle discretization of the flow and the Lagrangian advection scheme allow us to keep most of the advantages of the Lagrangian methods. The introduction of grid and remeshing process in vortex methods marks the birth of hybrid methods, with among them the Eulerian-Lagrangian decomposition domain methods and the Vortex-Particle-Mesh methods (like the Vortex-In-Cell and the semi-Lagrangian vortex methods). All these advances allowed by the hybrid approaches increased the interest and potential of VM in handling challenging CFD problems. Important efforts were therefore supplied to design efficient algorithms in the context of hybrid vortex methods and eventually gave birth to high performance scientific libraries, able to perform simulations of complex real flows arising in diverse application fields like hydrodynamics, turbulent wake dynamics, scalars transport or porous flows.

### 8.2. Future Research Perspectives

As described before, the academic research on vortex methods has brought multiple achievements on the accuracy, stability and applicability of this family of approaches. They have also been boosted with powerful high performance computing tools increasing their



capability to handle complex problems. Several perspectives appear to fill the future of the research related to this family of methods. The most current issue would be to strengthen their abilities for realistic and industrial applications. Below, some research perspectives are listed.

- Turbulent and high Reynolds number flow simulations. The recent researches highlight the road map for this target with two possibilities. The first option consists in proposing modern turbulence models for the velocity-vorticity filtered Navier-Stokes equations (i.e., LES-like approaches), based for example on those already developed by Mansfield et al. [148] on one side or Cocle et al. [151] on the other side. The second view, that takes its idea from the inherent nature of vortex methods, is the multi-resolution approach (based on the pioneer works of Cottet and Wray [147,157]) which consists in using two different levels of resolution for vorticity (fine level) and the velocity (coarse level). This multi-resolution would have a deep interest in problems where important Schmidt numbers are involved like in heat transfer (e.g., thermal conduction and convection [158]), passive-scalar transport [153] or sedimentation flows [141].
- Fluid Structure Interactions (FSI). The handling of no-slip boundary conditions has long been one of the main difficulties of vortex methods. The implementation of the Brinkman penalization method has significantly improved this drawback in the context of Vortex-Particle-Mesh methods. However, this approach is limited because of its low order and accuracy. The very recent introduction of immersed interface method (IIM) [131,132] and immersed lifting line approach (ILL) [12,99] introduced a real improvement and allows us to start a real progress in the challenging field of FSI.
- Environmental and industrial particular flows. Vortex methods belong to the family of particle methods with direct treatment of particles. Therefore, their application to problems like sediment transport [141,159] or internal multi-phase flows in industry with particle mixing, dispersion, deposition, and particle-wall collision [160], seems natural and deserves a special place in the future perspectives.
- Multi-phase flows. Besides the industrial fluid/particles two-phase flows evoked above, vortex methods also have a role to play in the handling of multi-phase flows in general, with for instance the development of models able to capture interfaces between two phases and to handle surface tension, variable fluid properties and high mass density-ratio flows. To the authors knowledge, only few works based on vortex methods have been dedicated to this topic [161].
- Finally, future vortex computations need to be continually strengthened with novel high performance computing parallelism techniques (hybrid multi-CPU/multi-GPU) and powerful algorithms (like the very recent 2D-3D Poisson solver released in 2021 by Caprace et al. [162]) in order to achieve more complex flow approximations.

**Author Contributions:** Conceptualization, C.M. and I.M.; investigation, C.M. and I.M.; writing-original draft preparation, C.M.; writing-review and editing, C.M. and I.M.; visualization, C.M. and I.M.; supervision, C.M. and I.M.; project administration, C.M. and I.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

Here is the list of the abbreviations used in this manuscript, in order of appearance:

CFD	Computational fluids dynamics
ODE	Ordinary differential equation
VM	Vortex methods
SPH	Smooth particle hydrodynamics

CFL	Courant-Friedricks-Levy
LCFL	Lagrangian Courant-Friedricks-Levy
RW	Random walk
PSE	Particle strength exchange
CSM	Core spreading method
DVM	Diffusion velocity method
CCF	Convolution of cut-off function
VRM	Vortex redistribution method
RBF	Radial basis functions
RPD	Regular point distributions
DVH	Diffused vortex hydrodynamics
VPM	Vortex-particle-mesh
VIC	Vortex-in-cell
P2M	Particle to Mesh
M2P	Mesh to Particle
FMM	Fast multiple method
DNS	Direct numerical simulation
LES	Large eddy simulation
BEM	Boundary elements method
IBM	Immersed boundary method
FFT	Fast Fourier Transform
GPU	Graphics processing unit
CPU	Central processing unit
AMR	Adaptive mesh refinement
FWT	Fast wavelet transform
LTS	Local time stepping
SGS	Sub-grid scale
SFS	Sub-filter scale
CMA-ES	Covariance Matrix Adaptation - Evolutionary Strategy
RL	Reinforcement learning
ILL	Immersed lifting line
HPC	High performance computing
FSI	Fluid structure interaction

## Nomenclature

in order of appearance

$t$	time
$x, y$	spacial locations
$\mathbf{Q}$	solution vector in given continuous system
$\mathbf{F}$	source term in given continuous system
$\Omega$	computational domain
$\mathbf{u}$	velocity field
$\delta$	Dirac distribution
$p$	particle
$\mathbf{v}_p$	volume of particle $p$ (denoted $v_p$ in 2D)
$\mathbf{x}_p$	location of particle $p$
$W_\varepsilon$	denomination of the smoothing radial symmetric function in Lagrangian methods
$\varepsilon$	radius of the smoothing radial symmetric function
$\Gamma$	circulation
$\times$	cross product
$\boldsymbol{\omega}$	vorticity field (denoted $\omega$ in 2D)
$\boldsymbol{\omega}_0$	initial vorticity field (denoted $\omega_0$ in 2D)
$(\cdot)^h$	quantity approximation
$\mathbf{u}_\infty$	far-field velocity
$\boldsymbol{\alpha}_p$	local circulation around $\mathbf{x}_p$ (denoted $\alpha_p$ in 2D)
$\star$	convolution product

<b>G</b>	Green's function
<b>K</b>	$\equiv \nabla \mathbf{G}$
$\zeta$	smooth cutoff function
$\zeta_\varepsilon$	smoothing radial symmetric function in Vortex blobs methods (based on $\zeta$ and on blob radius $\varepsilon$ )
$(\cdot)_\varepsilon$	quantity evaluated on mollified blob particles
$h$	particle spacing or grid spacing
$\Delta t$	time step
$\nu$	fluid viscosity
$Re$	Reynolds number. Also $Re_L, Re_c, Re_D$ and $Re_w$ for $Re$ based on <i>length, chord, diameter</i> and <i>width</i> respectively
$\eta_\varepsilon$	mollified diffusion kernel of given order, used in PSE schemes
$\mathbf{u}_d$	diffusion velocity
$\otimes$	tensor product
$\boldsymbol{\omega}_d$	viscous vorticity
$W$	interpolation/remeshing kernel
$S$	support of kernel $W$
$M_s$	half of grid points constituting the support of kernel $W$ ( $S = 2M_s$ )
$\Lambda_p$	ordinary interpolation/remeshing kernel of maximum order $p$
$M_p$	B-splines interpolation/remeshing kernel of maximum order $p$
$M'_4$	Monaghan interpolation/remeshing kernel
$\Lambda_{p,r}$	interpolation/remeshing kernel of regularity $r$ and maximum order $p$
$\phi_k$	basis functions in interpolation/remeshing process
$\mathbf{u}_s$	velocity of the solid immersed body
$\mathbf{x}_s$	any location on the solid surface
$\partial\boldsymbol{\omega}/\partial\mathbf{n}$	wall normal vorticity flux
$\mathcal{S}$	surface of the solid body
$\mathcal{S}_i$	vortex sheet element at the surface
$\alpha_i$	strength of vortex sheet element $\mathcal{S}_i$
$H$	Heaviside function
$\mathbf{s}$	tangent vector to the surface $\mathcal{S}$
$\kappa(s)$	surface curvature at point $s$
$(\mathbf{e}_r, \mathbf{e}_z, \mathbf{e}_\theta)$	basis vectors in cylindrical coordinate system
$\chi$	characteristic function of the solid body
$\lambda$	non-dimensionalized Brinkman penalization factor
$t_r, t_c$	threshold for refinement and compression, respectively, in wavelet-based AMR method
$\mathcal{F}_\Delta$	spatial filter $\mathcal{F}$ of size $\Delta$ in LES
$\boldsymbol{\tau}$	subgrid scale stress tensor in LES
$\nu_{sma}$	eddy viscosity in Smagorinsky model
$C$	coefficient in Smagorinsky model
$ \dot{\mathcal{S}} $	rate-of-strain tensor in Smagorinsky model
$(\cdot)$	low-pass filtered quantities in LES model
$R_{ij}$	subgrid scale vorticity stress in LES
$\mathbf{g} \equiv \partial R_{ij}/\partial x_j$	subfilter scale torque in LES
$Sc$	Schmidt number
$\kappa$	diffusivity of the advected scalar in transport problems.

## References

1. Gingold, R.; Monaghan, J. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* **1977**, *181*, 375–389. [[CrossRef](#)]
2. Monaghan, J. Why particle methods work. *SIAM J. Sci. Stat. Comput.* **1982**, *3*, 422–433. [[CrossRef](#)]
3. Monaghan, J. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* **1992**, *30*, 543–574. [[CrossRef](#)]
4. Shadloo, M.; Oger, G.; Le Touzé, D. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Comput. Fluids* **2016**, *136*, 11–34. [[CrossRef](#)]
5. Lind, S.; Rogers, B.; Stansby, P. Review of smoothed particle hydrodynamics: Towards converged Lagrangian flow modelling. *Proc. R. Soc. A* **2020**, *476*, 20190801. [[CrossRef](#)]

6. Gazzola, M.; Chatelain, P.; van Rees, W.M.; Koumoutsakos, P. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *J. Comput. Phys.* **2011**, *230*, 7093–7114. [[CrossRef](#)]
7. van Rees, W.; Gazzola, M.; Koumoutsakos, P. Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers. *J. Fluid Mech.* **2013**, *722*. [[CrossRef](#)]
8. Gazzola, M.; Hejazialhosseini, B.; Koumoutsakos, P. Reinforcement Learning and Wavelet Adapted Vortex Methods for Simulations of Self-propelled Swimmers. *SIAM J. Sci. Comput.* **2014**, *36*, B622–B639. [[CrossRef](#)]
9. Bernier, C.; Gazzola, M.; Ronsse, R.; Chatelain, P. Simulations of propelling and energy harvesting articulated bodies via vortex particle-mesh methods. *J. Comput. Phys.* **2019**, *392*, 34–55. [[CrossRef](#)]
10. Gazzola, M.; Mimeau, C.; Tchieu, A.; Koumoutsakos, P. Flow mediated interactions between two cylinders at finite Re numbers. *Phys. Fluids* **2012**, *24*, 043103. [[CrossRef](#)]
11. Parthasarathy, T.; Chan, F.; Gazzola, M. Streaming-enhanced flow-mediated transport. *J. Fluid Mech.* **2019**, *878*, 647–662. [[CrossRef](#)]
12. Caprace, D.G.; Chatelain, P.; Winckelmans, G. Wakes of rotorcraft in advancing flight: A large-eddy simulation study. *Phys. Fluids* **2020**, *32*, 087107. [[CrossRef](#)]
13. Stock, M.; Gharakhani, A.; Stone, C. Modeling rotor wakes with a hybrid OVERFLOW-vortex method on a GPU cluster. In Proceedings of the 28th AIAA Applied Aerodynamics Conference, Chicago, IL, USA, 28 June–1 July 2010; p. 4553.
14. Cogle, R.; Winckelmans, G.; Daeninck, G. Combining the Vortex-In-Cell and parallel Fast Multipole methods for efficient domain decomposition simulations. *J. Comput. Phys.* **2008**, *227*, 9091–9120. [[CrossRef](#)]
15. Chatelain, P.; Duponcheel, M.; Caprace, D.G.; Marichal, Y.; Winckelmans, G. Vortex particle-mesh simulations of vertical axis wind turbine flows: From the airfoil performance to the very far wake. *Wind Energy Sci.* **2017**, *2*, 317–328. [[CrossRef](#)]
16. Balty, P.; Caprace, D.G.; Waucquez, J.; Coquelet, M.; Chatelain, P. Multiphysics simulations of the dynamic and wakes of a floating Vertical Axis Wind Turbine. *J. Phys. Conf. Ser.* **2020**, *1618*, 062053. [[CrossRef](#)]
17. Huberson, S.; Rivoalen, E.; Voutsinas, S. Vortex particle methods in aeroacoustic calculations. *J. Comput. Phys.* **2008**, *227*, 9216–9240. [[CrossRef](#)]
18. Samarbakhsh, S.; Kornev, N. Simulation of a free circular jet using the vortex particle intensified LES ( $V\pi$ LES). *Int J. Heat Fluid Flow* **2019**, *80*, 108489. [[CrossRef](#)]
19. Poncet, P.; Hildebrand, R.; Cottet, G.H.; Koumoutsakos, P. Spatially distributed control for optimal drag reduction of the flow past a circular cylinder. *J. Fluid Mech.* **2008**, *599*, 111–120. [[CrossRef](#)]
20. Mimeau, C.; Mortazavi, I.; Cottet, G.H. Passive control of the flow around a hemisphere using porous media. *Eur. J. Mech. B Fluids* **2017**, *65*, 213–226. [[CrossRef](#)]
21. Gustafson, K.; Sethian, J. (Eds.) *Vortex Methods and Vortex Motion*; SIAM: Philadelphia, PA, USA, 1991.
22. Cottet, G.H.; Koumoutsakos, P. *Vortex Methods—Theory and Practice*; Cambridge University Press: Cambridge, UK, 2000.
23. Yokota, R.; Obi, S. Vortex Methods for the Simulation of Turbulent Flows: Review. *J. Fluid Sci. Technol.* **2011**, *6*, 14–29. [[CrossRef](#)]
24. Mimeau, C. Conception and Implementation of a Hybrid Vortex Penalization Method for sOlid-Fluid-Porous Media: Application to the Passive Control of Incompressible Flows. Ph.D. Thesis, Université de Grenoble, Grenoble, France, 2015.
25. Prager, W. Die Druckverteilung an Korpern in ebener Potentialstromung. *Phys. Z.* **1928**, *29*, 865–869.
26. Rosenhead, L. The Formation of vortices from a surface of discontinuity. *Proc. R. Soc. Lond. Ser. A* **1931**, *134*, 170–192.
27. Chorin, A.J.; Bernard, P.S. Discretization of a Vortex Sheet, with an Example of Roll-Up. *J. Comput. Phys.* **1973**, *13*, 423–429. [[CrossRef](#)]
28. Anderson, C.; Greengard, C. On vortex methods. *SIAM J. Numer. Anal.* **1985**, *22*, 413–440. [[CrossRef](#)]
29. Cottet, G.H. A new approach for the analysis of Vortex Methods in two and three dimensions. *Annales de l'I. H. P. Analyse non Linéaire* **1988**, *5*, 227–285. [[CrossRef](#)]
30. Leonard, A. Numerical simulation of interacting vortex filaments. In Proceedings of the IV International Conference on Numerical Methods of Fluid Dynamics, Boulder, CO, USA, 24–28 June 1974; Springer: New York, NY, USA, 1975.
31. Leonard, A. Vortex methods for flow simulation. *J. Comput. Phys.* **1980**, *37*, 289–335. [[CrossRef](#)]
32. Leonard, A. Computing three-dimensional incompressible flows with vortex elements. *Ann. Rev. Fluid Mech.* **1985**, *17*, 523–559. [[CrossRef](#)]
33. Angelidis, A.; Neyret, F. Simulation of smoke based on vortex filament primitives. In Proceedings of the ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA), Los Angeles, CA, USA, 29–31 July 2005.
34. Hald, O. Convergence of vortex methods II. *SIAM J. Numer. Anal.* **1979**, *16*, 726–755. [[CrossRef](#)]
35. Beale, J.; Majda, A. Vortex methods II: High order accuracy in 2 and 3 dimensions. *Math. Comput.* **1982**, *32*, 29–52.
36. Knio, O.; Ghoniem, A.F. Numerical study of a three-dimensional vortex method. *J. Comput. Phys.* **1990**, *86*, 75–106. [[CrossRef](#)]
37. Chorin, A.J. Numerical study of slightly viscous flow. *J. Fluid Mech.* **1973**, *57*, 785–796. [[CrossRef](#)]
38. Prandtl, W. Über Flüssigkeitsbewegung bei sehr kleiner Reibung. In Proceedings of the Fourth Mathematics Congress, Heidelberg, Germany, 8–13 August 1904; pp. 484–491.
39. Beale, J.T.; Majda, A. Rates of convergence for viscous splitting of the Navier-Stokes equations. *Math. Comput.* **1981**, *37*, 243–259. [[CrossRef](#)]
40. Marchioro, C.; Pulvirenti, M. Hydrodynamics in two dimensions and vortex theory. *Commun. Math. Phys.* **1982**, *84*, 483–503. [[CrossRef](#)]

41. Goodman, J. Convergence of the random vortex method. *Commun. Pure Appl. Math.* **1987**, *40*, 189–220. [[CrossRef](#)]
42. Long, D.G. Convergence of the random vortex method in two dimensions. *J. Am. Math. Soc.* **1988**, *1*, 779–804. [[CrossRef](#)]
43. Chorin, A.J. *Computational Fluid Mechanics*; Academic Press: New York, NY, USA, 1989.
44. van Dommelen, L.L. Computation of unsteady separation using Lagrangian procedures. In Proceedings of the IUTAM Symposium on Boundary Layer Separation, London, UK, 26–28 August 1986; pp. 73–87.
45. Smith, P.; Stansby, P. An efficient surface algorithm for random-particle simulation of vorticity and heat transport. *J. Comput. Phys.* **1989**, *81*, 349–371. [[CrossRef](#)]
46. Slaouti, A.; Stansby, P. Flow around two circular cylinders by the random-vortex method. *J. Fluids Struct.* **1992**, *6*, 641–670. [[CrossRef](#)]
47. Wang, S.C. Control of Dynamic Stall. Ph.D. Thesis, FAMU-FSU, College of Engineering, Tallahassee, FL, USA, 1995.
48. Lewis, R.I. *Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems*; Cambridge University Press: Cambridge, UK, 1990.
49. Milinazzo, F.; Saffman, P.G. The calculation of large Reynolds number two-dimensional flow using discrete vortices with random walk. *J. Comput. Phys.* **1977**, *23*, 380–392. [[CrossRef](#)]
50. Gharakhani, A.; Ghoniem, A. Three-Dimensional Vortex Simulation of Time Dependent Incompressible Internal Viscous Flows. *J. Comput. Phys.* **1997**, *134*, 75–95. [[CrossRef](#)]
51. Degond, P.; Mas-Gallic, S. The weighted particle method for convection-diffusion equations. *Math. Comput.* **1989**, *53*, 485–526.
52. Cottet, G.H.; Mas-Gallic, S. Une méthode de décomposition pour une équation de type convection-diffusion combinant résolution explicite et méthode particulière. *C. R. Acad. Sci. Paris* **1983**, *297*, 133–136.
53. Cottet, G.H.; Mas-Gallic, S. A particle method to solve the Navier-Stokes system. *Numer. Math.* **1990**, *57*, 1–23. [[CrossRef](#)]
54. Koumoutsakos, P.; Leonard, A. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.* **1995**, *296*, 1–38. [[CrossRef](#)]
55. Ploumhans, P.; Winckelmans, G.S.; Salmon, J.K.; Leonard, A.; Warren, M.S. Vortex Methods for Direct Numerical Simulation of Three-Dimensional Bluff Body Flows: Applications to the Sphere at  $Re = 300, 500$  and  $1000$ . *J. Comput. Phys.* **2002**, *178*, 427–463. [[CrossRef](#)]
56. Yokota, R.; Sheel, T.; Obi, S. Calculation of isotropic turbulence using a pure Lagrangian vortex method. *J. Comput. Phys.* **2007**, *226*, 1589–1606. [[CrossRef](#)]
57. Kuwahara, K.; Takami, H. Numerical studies of two-dimensional vortex motion by a system of point vortices. *J. Phys. Soc. Jpn.* **1973**, *34*, 247–253. [[CrossRef](#)]
58. Greengard, C. The core spreading vortex method approximates the wrong equation. *J. Comput. Phys.* **1985**, *61*, 345–348. [[CrossRef](#)]
59. Rossi, L.F. Resurrecting core spreading vortex methods: A new scheme that is both deterministic and convergent. *SIAM J. Sci. Stat. Comput.* **1996**, *17*, 370–397. [[CrossRef](#)]
60. Barba, L.; Leonard, A.; Allen, C. Advances in viscous vortex methods—Meshless spatial adaption based on radial basis function interpolation. *Int. J. Numer. Methods Fluids* **2005**, *47*, 387–421. [[CrossRef](#)]
61. Fronteau, J.; Combis, P. A lie admissible method of integration of Fokker-Planck equations with non linear coefficients (exact and numerical solutions). *Hadronic J.* **1984**, *7*, 911–930.
62. Ogami, Y.; Akamatsu, T. Viscous flow simulation using the discrete vortex model—The diffusion velocity model. *Comput. Fluids* **1991**, *19*, 433–441. [[CrossRef](#)]
63. Beaudoin, A.; Huberson, S.; Rivoalen, E. Simulation of anisotropic diffusion by means of a diffusion velocity method. *J. Comput. Phys.* **2003**, *186*, 122–135. [[CrossRef](#)]
64. Clarke, N.; Tutty, O. Construction and validation of a discrete vortex method for the two-dimensional incompressible Navier-Stokes equations. *Comput. Fluids* **1994**, *23*, 751–783. [[CrossRef](#)]
65. Lucchesi, M.; Allouch, S.; Le Maitre, O.; Mustapha, K.; Knio, O. Particle simulation of space-fractional diffusion equations. *Comput. Part. Mech.* **2020**, *7*, 491–507. [[CrossRef](#)]
66. Rivoalen, E.; Huberson, S.; Hauville, F. Simulation numérique des équations de Navier-Stokes 3D par une méthode particulière. *C. R. Acad. Sci. Ser. IIB Mech. Phys. Chem. Astron.* **1997**, *324*, 543–549. [[CrossRef](#)]
67. Fishelov, D. A new vortex scheme for viscous flows. *J. Comput. Phys.* **1990**, *86*, 211–224. [[CrossRef](#)]
68. Shankar, S.; van Dommelen, L. A new diffusion procedure for vortex methods. *J. Comput. Phys.* **1996**, *127*, 88–109. [[CrossRef](#)]
69. Barba, L. Vortex Method for Computing High-Reynolds Number Flows: Increased Accuracy with a Fully Mesh-Less Formulation. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 2004.
70. Rossinelli, D.; Koumoutsakos, P. Vortex methods for incompressible flow simulations on the GPU. *Vis. Comput.* **2008**, *24*, 699–708. [[CrossRef](#)]
71. Henshaw, W.; Kreiss, H.O.; Reyna, L. On the smallest scale for the compressible Navier-Stokes equations. *Theor. Comput. Fluid Dyn.* **1989**, *1*, 65–95.
72. Krasny, R. A study of singularity formation in a vortex sheet by the point vortex approximation. *J. Fluid Mech.* **1986**, *167*, 65–93. [[CrossRef](#)]
73. Beale, J.T. On the accuracy of vortex methods at large times. In *Computational Fluid Dynamics and Reacting Gas Flows*; Engquist, B., Majda, A., Luskin, M., Eds.; Springer: New York, NY, USA, 1988; pp. 19–32.



74. Choquin, J.P.; Lucquin-Desreux, B. Accuracy of a deterministic particle method for the Navier-Stokes equations. *Int. J. Numer. Fluids* **1988**, *8*, 1439–1458. [[CrossRef](#)]
75. Russo, G.; Strain, J. Fast triangulated vortex methods for the 2D Euler equations. *J. Comput. Phys.* **1994**, *111*, 291–323. [[CrossRef](#)]
76. Strain, J. Fast adaptive 2D vortex methods. *J. Comput. Phys.* **1997**, *132*, 108–122. [[CrossRef](#)]
77. Schrader, B.; Sbalzarini, I.; Redoux, S. Discretization correction of general integral PSE operators for particle methods. *J. Comput. Phys.* **2010**, *229*, 4159–4182. [[CrossRef](#)]
78. Eldredge, J.; Colonius, T.; Leonard, A. A general deterministic treatment of derivatives in particle methods. *J. Comput. Phys.* **2002**, *180*, 686–709. [[CrossRef](#)]
79. Rossi, E.; Colagrossi, A.; Bouscasse, B.; Graziani, G. The Diffused Vortex Hydrodynamics Method. *Commun. Comput. Phys.* **2015**, *18*, 351–379. [[CrossRef](#)]
80. Huberson, S.; Jollès, A. Correction de l'erreur de projection dans les méthodes particules/maillage. *La Recherche Aéronautique* **1990**, *4*, 1–6.
81. Cottet, G.H.; Magni, A. TVD remeshing schemes for particle methods. *C. R. Acad. Sci. Paris* **2009**, *347*, 1367–1372. [[CrossRef](#)]
82. Magni, A.; Cottet, G.H. Accurate, non-oscillatory remeshing schemes for particle methods. *J. Comput. Phys.* **2012**, *231*, 152–172. [[CrossRef](#)]
83. Schoenberg, I.J. Contributions to the problem of approximation of equidistant data by analytic functions. *Q. Appl. Math.* **1946**, *4*, 112–141. [[CrossRef](#)]
84. Magni, A. Méthodes Particulières Avec Remaillage: Analyse Numérique, Nouveaux Schémas Et Applications Pour La Simulation D'équations De Transport. Ph.D. Thesis, Université de Grenoble, Grenoble, France, 2011.
85. Monaghan, J. Extrapolating B-splines for interpolation. *J. Comput. Phys.* **1985**, *60*, 253–262. [[CrossRef](#)]
86. Gazzola, M.; van Rees, W.M.; Koumoutsakos, P. C-start: Optimal start of larval fish. *J. Fluid Mech.* **2012**, *698*, 5–18. [[CrossRef](#)]
87. Bergdorf, M.; Koumoutsakos, P. A Lagrangian particle-wavelet method. *SIAM Multiscale Model. Simul.* **2006**, *5*, 980–995. [[CrossRef](#)]
88. Cottet, G.H.; Etancelin, J.M.; Perignon, F.; Picard, C. High order Semi-Lagrangian particles for transport equations: Numerical analysis and implementation issues. *ESAIM Math. Model. Numer. Anal.* **2014**, *48*, 1029–1060. [[CrossRef](#)]
89. Cottet, G.H. Particle-grid domain decomposition methods for the Navier-Stokes equations in exterior domains. *Lect. Appl. Math. Ser. Am. Math. Soc. N. Y.* **1991**, *28*, 100–118.
90. Cottet, G.H.; Koumoutsakos, P.; Ould-Salihi, M. Vortex Methods with Spatially Varying Cores. *J. Comput. Phys.* **2000**, *162*, 164–185. [[CrossRef](#)]
91. Huberson, S.; Voutsinas, S. Particles and grid. *Comput. Fluids* **2002**, *31*, 607–625. [[CrossRef](#)]
92. Guermont, J.L.; Huberson, S.; Shen, W.Z. Simulation of 2D external viscous flows by means of a domain decomposition method. *J. Comput. Phys.* **1993**, *108*, 343–352. [[CrossRef](#)]
93. Christiansen, J. Numerical solution of hydrodynamics by the method of point vortices. *J. Comput. Phys.* **1973**, *13*, 363–379. [[CrossRef](#)]
94. Mortazavi, I.; Giovannini, A. The simulation of vortex dynamics downstream of a plate separator using a vortex-finite element method. *Int. J. Fluid Dyn.* **2001**, *5*, 31–48.
95. Ould-Salihi, M.; Cottet, G.H.; El Hamraoui, M. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM J. Sci. Comp.* **2000**, *22*, 1655–1674. [[CrossRef](#)]
96. Gallizio, F. Analytical and Numerical Vortex Methods to Model Separated Flows. Ph.D. Thesis, Politecnico di Torino, Turin, Italy, 2009.
97. van Rees, W.M.; Leonard, A.; Pullin, D.; Koumoutsakos, P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *J. Comput. Phys.* **2011**, *230*, 2794–2805. [[CrossRef](#)]
98. Mimeau, C.; Gallizio, F.; Cottet, G.H.; Mortazavi, I. Vortex penalization method for bluff body flows. *Int. J. Numer. Meth. Fluids* **2015**, *79*, 55–83. [[CrossRef](#)]
99. Caprace, D.G.; Winckelmans, G.; Chatelain, P. An immersed lifting and dragging line model for the Vortex Particle-Mesh method. *Theor. Comput. Fluid Dyn.* **2020**, *34*, 21–48. [[CrossRef](#)]
100. Mimeau, C.; Cottet, G.H.; Mortazavi, I. Direct numerical simulations of three-dimensional flows past obstacles with a vortex penalization method. *Comput. Fluids* **2016**, *136*, 331–347. [[CrossRef](#)]
101. Morgenthal, G.; Walther, J.H. An immersed interface method for the Vortex-In-Cell algorithm. *Comput. Struct.* **2007**, *85*, 712–726. [[CrossRef](#)]
102. Kudela, H.; Kozłowski, T. Vortex-in-cell method for exterior problems. *J. Theor. Appl. Mech.* **2009**, *47*, 779–796.
103. Cottet, G.H.; Poncet, P. Simulation and control of three-dimensional wakes. *Comput. Fluids* **2004**, *33*, 697–713. [[CrossRef](#)]
104. Creusé, E.; Giovannini, A.; Mortazavi, I. Vortex simulation of active control strategies for transitional backward-facing step flows. *Comput. Fluids* **2009**, *38*, 1348–1360. [[CrossRef](#)]
105. Kornev, N. Hybrid method based on embedded coupled simulation of vortex particles in grid based solution. *Comput. Part. Mech.* **2018**, *5*, 269–283. [[CrossRef](#)]
106. Chorin, A.J. Vortex sheet approximation of boundary layers. *J. Comput. Phys.* **1978**, *27*, 428–442. [[CrossRef](#)]
107. Smith, P.; Stansby, P. Impulsively started flow around a circular cylinder by the vortex method. *J. Fluid Mech.* **1988**, *194*, 45–77. [[CrossRef](#)]



108. Sethian, J.A.; Ghoniem, A.F. Validation Study of Vortex Methods. *J. Comput. Phys.* **1988**, *74*, 283–317. [[CrossRef](#)]
109. Cheer, A.J. Numerical study of incompressible slightly viscous flow past blunt bodies and airfoils. *SIAM J. Sci. Stat. Comput.* **1983**, *4*, 685–705. [[CrossRef](#)]
110. Mortazavi, I.; Micheau, P.; Giovannini, A. Numerical convergence of vortex method for a high Reynolds number 2D bluff-body flow. *C. R. Mec.* **2002**, *330*, 409–416. [[CrossRef](#)]
111. Chorin, A.J. Vortex models and boundary layer instability. *SIAM J. Sci. Stat. Comput.* **1980**, *1*, 1–21. [[CrossRef](#)]
112. Fishelov, D. Vortex Methods for Slightly Viscous Three Dimensional Flow. *SIAM J. Sci. Stat. Comput.* **1990**, *3*, 399–424. [[CrossRef](#)]
113. Gharakhani, A.; Ghoniem, A. Simulation of the piston driven flow inside a cylinder with an eccentric port. *J. Fluids Eng.* **1998**, *120*, 319–326. [[CrossRef](#)]
114. Anderson, C.; Reider, M. Investigation of the use of Prandtl/Navier-Stokes Equation Procedures for Two-Dimensional Incompressible Flows. *Phys. Fluids* **1994**, *6*, 2380–2389. [[CrossRef](#)]
115. Walther, J.H.; Larsen, A. Two dimensional discrete vortex method for application to bluff body aerodynamics. *J. Wind Eng. Ind. Aerodyn.* **1997**, *67–68*, 183–193.
116. Cottet, G.H. A vorticity creation algorithm for the Navier-Stokes equations in arbitrary domain. In *Navier-Stokes Equations and Related Non-Linear Problems*; Sequeira, A., Ed.; Plenum Press Publishers: New York, NY, USA, 1994.
117. Koumoutsakos, P.; Leonard, A.; Pépin, F. Boundary conditions for viscous vortex methods. *J. Comput. Phys.* **1994**, *113*, 52–61. [[CrossRef](#)]
118. Casciola, C.M.; Piva, R.; Bassanini, P. Vorticity generation on a flat surface in 3D flows. *J. Comput. Phys.* **1996**, *129*, 345–356. [[CrossRef](#)]
119. Poncet, P. Méthodes Particulières Pour la Simulation des Sillages Tridimensionnels. Ph.D. Thesis, Université Joseph Fourier–Grenoble I, Grenoble, France, 2001.
120. Cottet, G.H.; Poncet, P. Advances in direct numerical simulation of 3D wall-bounded flows by Vortex-In-Cell methods. *J. Comput. Phys.* **2003**, *193*, 136–158. [[CrossRef](#)]
121. Cooper, C.; Barba, L. Panel-free boundary conditions for viscous vortex methods. In Proceedings of the AIAA Computational Fluid Dynamics Conference, San Antonio, TX, USA, 22–25 June 2009.
122. Ploumhans, P.; Winckelmans, G.S. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *J. Comput. Phys.* **2000**, *165*, 354–406. [[CrossRef](#)]
123. Poncet, P. Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation. *J. Comput. Phys.* **2009**, *228*, 7268–7288. [[CrossRef](#)]
124. Caltagirone, J.P. Sur l’interaction fluide-milieu poreux: Application au calcul des efforts exercés sur un obstacle par un fluide visqueux. *C. R. Acad. Sci. Paris* **1994**, *318*, 571–577.
125. Angot, P.; Bruneau, C.H.; Fabrie, P. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.* **1999**, *81*, 497–520. [[CrossRef](#)]
126. Kevlahan, N.; Ghidaglia, J.M. Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization. *Eur. J. Mech.* **2001**, *20*, 333–350. [[CrossRef](#)]
127. Morency, F.; Beaugendre, H.; Gallizio, F. Aerodynamic force evaluation for ice shedding phenomenon using vortex in cell scheme, penalisation and level set approaches. *Int. J. Comput. Fluid Dyn.* **2012**, *26*, 435–450. [[CrossRef](#)]
128. Coquerelle, M.; Cottet, G.H. A vortex level-set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.* **2008**, *227*, 9121–9137. [[CrossRef](#)]
129. Hejlesen, M.; Koumoutsakos, P.; Leonard, A.; Walther, J. Iterative Brinkman penalization for remeshed vortex methods. *J. Comput. Phys.* **2015**, *280*, 547–562. [[CrossRef](#)]
130. Gillis, T.; Winckelmans, G.; Chatelain, P. An efficient iterative penalization method using recycled Krylov subspaces and its application to impulsively started flows. *J. Comput. Phys.* **2017**, *347*, 490–505. [[CrossRef](#)]
131. Marichal, Y.; Chatelain, P.; Winckelmans, G. Immersed interface interpolation schemes for particle–mesh methods. *J. Comput. Phys.* **2016**, *326*, 947–972. [[CrossRef](#)]
132. Gillis, T.; Winckelmans, G.; Chatelain, P. A 2D immersed interface Vortex Particle-Mesh method. *J. Comput. Phys.* **2019**, *394*, 700–718. [[CrossRef](#)]
133. LeVeque, R.J.; Li, Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* **1994**, *31*, 1019–1044. [[CrossRef](#)]
134. Rossinelli, D.; Hejzialisosseini, B.; van Rees, W.M.; Gazzola, M.; Bergdorf, M.; Koumoutsakos, P. MRAG-I2D: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *J. Comput. Phys.* **2015**, *288*, 1–18. [[CrossRef](#)]
135. Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348. [[CrossRef](#)]
136. Marzouk, Y.M.; Ghoniem, A.F. K-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical N-body simulations. *J. Comput. Phys.* **2005**, *207*, 493–528. [[CrossRef](#)]
137. Yokota, R.; Narumi, T.; Sakamaki, R.; Kameoka, S.; Obi, S.; Yasuoka, K. Fast multipole methods on a cluster of GPUs for the meshless simulation of turbulence. *Comput. Phys. Commun.* **2009**, *180*, 2066–2078. [[CrossRef](#)]
138. Giannopoulou, O.; Colagrossi, A.; Di Mascio, A.; Mascia, C. Chorin’s approaches revisited: Vortex Particle Method vs. Finite Volume Method. *Eng. Anal. Bound. Elem.* **2019**, *106*, 371–388. [[CrossRef](#)]

139. Hu, Q.; Gumerov, N.; Yokota, R.; Barba, L.A. Duraiswami, R. Scalable fast multipole methods for vortex element methods. In Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, Salt Lake City, UT, USA, 10–16 November 2012; pp. 1408–1409.
140. Rossinelli, D.; Bergdorf, M.; Cottet, G.H.; Koumoutsakos, P. GPU accelerated simulations of bluff body flows using vortex particle methods. *J. Comput. Phys.* **2010**, *229*, 3316–3333. [[CrossRef](#)]
141. Keck, J.B. Numerical Modelling and High Performance Computing for Sediment Flows. Ph.D. Thesis, Université Grenoble Alpes, Grenoble, France, 2019.
142. Yokota, R.; Barba, L.A.; Narumi, T.; Yasuoka, K. Petascale turbulence simulation using a highly parallel fast multipole method on GPUs. *Comput. Phys. Commun.* **2013**, *184*, 445–455. [[CrossRef](#)]
143. Bergdorf, M.; Cottet, G.H.; Koumoutsakos, P. Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Model. Sim.* **2005**, *4*, 328–357. [[CrossRef](#)]
144. Rasmussen, J.T.; Cottet, G.H.; Walther, J.H. A multiresolution remeshed Vortex-In-Cell algorithm using patches. *J. Comput. Phys.* **2011**, *230*, 6742–6755. [[CrossRef](#)]
145. El Ossmani, M.; Poncet, P. Efficiency of Multiscale Hybrid Grid-Particle Vortex Methods. *Multiscale Model. Simul.* **2010**, *8*, 1671–1690. [[CrossRef](#)]
146. Sagaut, P.; Deck, S.; Terracol, M. *Multiscale and Multiresolution Approaches in Turbulence. LES, DES and Hybrid RANS/LES Methods: Applications and Guidelines*; Imperial College Press: London, UK, 2013.
147. Cottet, G.H. Artificial Viscosity Models for Vortex and Particle Methods. *J. Comput. Phys.* **1996**, *127*, 199–208. [[CrossRef](#)]
148. Mansfield, J.; Knio, O.; Meneveau, C. A Dynamic LES Scheme for the Vorticity Transport Equation: Formulation and a Priori Tests. *J. Comput. Phys.* **1998**, *145*, 693–730. [[CrossRef](#)]
149. Mansfield, J.; Knio, O.; Meneveau, C. Dynamic LES of Colliding Vortex Rings Using a 3D Vortex Method. *J. Comput. Phys.* **1999**, *152*, 305–745. [[CrossRef](#)]
150. Pinon, G.; Bratec, H.; Huberson, S.; Pignot, G.; Rivoalen, E. Vortex method for simulation of a 3D round jet in a cross-stream. *J. Turbul.* **2005**, *6*, N18. [[CrossRef](#)]
151. Cocle, R.; Dufresne, L.; Winckelmans, G. Investigation of multiscale subgrid scale models for LES of instabilities and turbulence in wake vortex systems. In *Complex Effects in Large Eddy Simulations, Lecture Notes in Computational Science and Engineering (LNCSE)*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 56.
152. Novati, G.; Verma, S.; Alexeev, D.; Rossinelli, D.; van Rees, W.; Koumoutsakos, P. Synchronisation through learning for two self-propelled swimmers. *Bioinspir. Biomim.* **2017**, *12*, 036001. [[CrossRef](#)]
153. Etancelin, J.M.; Cottet, G.H.; Pérignon, F.; Picard, C. Multi-CPU and multi-GPU hybrid computations of multi-scale scalar transport. In Proceedings of the 26th International Conference on Parallel Computational Fluid Dynamics, Trondheim, Norway, 20–22 May 2014; pp. 83–84.
154. Etancelin, J.M. Couplage de Modèles, Algorithmes Multi-échelles et Calcul Hybride. Ph.D. Thesis, Université de Grenoble, Grenoble, France, 2014.
155. Mimeau, C.; Mortazavi, I.; Cottet, G.H. Passive Flow Control Around a Semi-Circular Cylinder Using Porous Coatings. *Int. J. Flow Control* **2014**, *6*, 43–60.
156. Etancelin, J.M.; Moonen, P.; Poncet, P. Improvement of remeshed Lagrangian methods for the simulation of dissolution processes at pore-scale. *Adv. Water Resour.* **2020**, *146*, 103780. [[CrossRef](#)]
157. Cottet, G.H.; Wray, A. Anisotropic grid-based formulas for subgrid-scale models. In *Annual Research Briefs*; Center for Turbulence Research: Stanford, CA, USA, 1997.
158. Ostilla-Monico, R.; Yang, Y.; van der Poel, E.; Lohse, D.; Verzicco, R. A multiple-resolution strategy for Direct Numerical Simulation of scalar turbulence. *J. Comput. Phys.* **2015**, *301*, 308–321. [[CrossRef](#)]
159. Keck, J.B.; Cottet, G.H.; Meiburg, E.; Mortazavi, I.; Picard, C. Double-diffusive sedimentation at high Schmidt numbers: Semi-Lagrangian simulations. *Phys. Rev. Fluids* **2021**, to appear.
160. Iso, Y.; Kamemoto, K. A Grid-Free Lagrangian Approach of Vortex Method and Particle Trajectory Tracking Method Applied to Internal Fluid-Solid Two-Phase Flows. *J. Fluids Eng.* **2008**, *130*, 011401. [[CrossRef](#)]
161. Lorieul, G. Development and Validation of a 2D Vortex-Particle-Mesh Method for Incompressible Multi-Phase Flows. Ph.D. Thesis, Université Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, 2018.
162. Caprace, D.G.; Gillis, T.; Chatelain, P. FLUPS—A Fourier-based Library of Unbounded Poisson Solvers. *SIAM J. Sci. Comput.* **2021**, *43*, C31–C60. [[CrossRef](#)]