

An immersed interface method for the Vortex-In-Cell algorithm

G. Morgenthal ^{a,*,1}, J.H. Walther ^{b,2}

^a *Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom*

^b *Department of Mechanical Engineering, Fluid Mechanics, Technical University of Denmark, Building 403, DK-2800 Lyngby, Denmark*

Received 2 June 2006; accepted 8 January 2007

Available online 12 March 2007

Abstract

The paper presents a two-dimensional immersed interface technique for the Vortex-In-Cell (VIC) method for simulation of flows past bodies of complex geometry. The particle–mesh VIC algorithm is augmented by a local particle–particle correction term in a Particle–Particle–Mesh (P³M) context to resolve sub-grid scales incurred by the presence of the immersed interface. The particle–particle correction furthermore allows to disjoin mesh and particle resolution by explicitly resolving sub-grid scales on the particles. This P³M algorithm uses an influence matrix technique to annihilate the anisotropic sub-grid scales and adds an exact particle–particle correction term. Free-space boundary conditions are satisfied through the use of modified Green’s functions in the solution of the Poisson equation for the streamfunction. The concept is extended such as to provide exact velocity predictions on the mesh with free-space boundary conditions.

The random walk technique is employed for the diffusion in order to relax the need for a remeshing of the computational elements close to solid boundaries. A novel partial remeshing technique is introduced which only performs remeshing of the vortex elements which are located sufficiently distant from the immersed interfaces, thus maintaining a sufficient spatial representation of the vorticity field.

Convergence of the present P³M algorithm is demonstrated for a circular patch of vorticity. The immersed interface technique is applied to the flow past a circular cylinder at a Reynolds number of 3000 and the convergence of the method is demonstrated by a systematic refinement of the spatial parameters. Finally, the flow past a cactus-like geometry is considered to demonstrate the efficient handling of complex bluff body geometries. The simulations offer an insight into physically interesting flow behavior involving a temporarily negative total drag force on the section.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Vortex methods; Lagrangian method; Immersed boundaries; Particle–mesh algorithm; Random walk; Remeshing

1. Introduction

Vortex methods have in the past proven valuable numerical tools in the prediction of complex unsteady flows due to their robustness and lack of time consuming mesh generation required in traditional Eulerian methods cf. [1–14]. They utilize a discretization of the vorticity field

by particles and a Lagrangian formulation of the governing Navier–Stokes equations to determine their evolution. The major advantages of the classical vortex method over grid-based methods are an automatic adaptivity of the computational elements and low numerical dissipation. However, in terms of computational cost, the method is rendered impractical for high-resolution simulations due to the N -body problem involving the mutual interaction of all N_p vortices for the calculation of the fluid velocity field. This leads to a cost of $\mathcal{O}(N_p^2)$ and limits the practically usable number of computational elements. Fast multipole methods have been developed to reduce the computational cost to $\mathcal{O}(N_p \log N_p)$ [15] and $\mathcal{O}(N_p)$ [16]. For problems with simple geometries, alternative hybrid particle–mesh algorithms

* Corresponding author.

E-mail address: guido@morgenthal.org (G. Morgenthal).

¹ Present address: 4A Village Tower, 7 Village Road, Happy Valley, Hong Kong SAR, China.

² Also at: Institute of Computational Science, ETH Zürich, 8092 Zürich, Switzerland.

such as the Vortex-In-Cell (VIC) method offer a computational cost of $\mathcal{O}(N_p \log N_p)$ or $\mathcal{O}(N_p)$ by employing fast FFT or iterative solvers for the field equation on a grid. High-order moment conserving interpolation kernels [17,18] are used for the projection of the vorticity field from the particles to the mesh and the interpolation of the fluid velocity to the particles to retain the accuracy of the method.

Immersed interface methods utilize fast Poisson solvers on a regular mesh and enforce the appropriate boundary conditions on immersed interfaces through an additional forcing of the dynamics of the flow, e.g. through body forces [19–24]. The enforcement of the boundary condition on the immersed interface is reduced to determining the proper forcing which is typically computed iteratively during the time stepping procedure [25]. The immersed interface techniques in vortex methods enjoy the advantage of a clear separation of dynamics and kinematics, ensured by the velocity–vorticity formulation. Moreover, the dynamics is computed on the particle, thus removing the Courant criterion traditionally limiting the time step in Eulerian methods.

The present work uses a novel particle–particle–mesh (P³M) immersed interface method for particle methods in the framework of the Vortex-In-Cell method with the following features: (i) efficient solution of the Poisson equation using a fast FFT solver, (ii) exact prescription of free-space boundary conditions using a minimum number of grid points, (iii) automatic resolution of sub-grid scales through the application of direct particle–particle interaction corrections in the near field, (iv) whilst we present a two-dimensional implementation the method is readily extendible to three dimensions (3D). However, in 3D to ensure a divergence-free vorticity field we expect to replace the random walk technique with, e.g., the Particle-Strength-Exchange (PSE) scheme [26]. The proposed P³M algorithm replaces tree-based algorithms such as the fast multipole method by computing the interaction of distant particles on the mesh and the interaction of particles in close proximity through an influence matrix technique and an exact particle–particle correction term.

The convergence of vortex methods requires an occasional re-initialization of the particles to ensure the spatial representation of the fields particularly in regions of significant strain in the flow [27,28]. This re-meshing is straightforwardly implemented in a VIC scheme through the use of the VIC particle–mesh moment conserving interpolation formulae, but special treatment is required near the immersed interfaces to avoid the creation of fluid particles within the solid region. For bodies of regular shape the geometric properties can be exploited [7] but for irregular bodies the conservation of the statistical moments is non-trivial and one-sided formulae are usually applied, cf. Ploumhans and Winckelmans [12]. Further need for a regularized particle pattern arises from the use of diffusion schemes like the PSE by Degond and Mas-Gallic [29] which require a continued remeshing throughout the domain to (i) secure a regular particle map and (ii) obtain

a mechanism for creating particles at the boundaries of the vorticity field. In this study the random walk method [30] is utilized for the modelling of the diffusion term of the Navier–Stokes equation. This method has been shown to provide sufficiently accurate results for engineering applications, e.g. in bridge aerodynamics [31]. Moreover, it is substantially less sensitive to the particle layout. Use of this fact is made in regions adjacent to the body surface and a partial remeshing strategy is proposed which refrains from replacing particles that would, according to the interpolation kernel, create particles inside the solid body. This ensures a sufficient vorticity support and uniform particle spacing in most of the domain but relies on a particle creation strategy consistent with the random walk method used herein.

The paper is organized as follows: The governing equations are described in Section 2. The classical vortex method is outlined in Section 3, and the proposed VIC immersed interface method is described in Section 4. The partial remeshing technique is described in Section 5. Section 6 presents the results obtained with the present algorithm including the study of a patch of vorticity to demonstrate the convergence of the P³M algorithm (Section 6.1). The convergence of the immersed interface method is presented in Section 6.2 for the flow past a circular cylinder at moderate Reynolds number, and the impulsively started flow past a cactus-like geometry is presented in Section 6.3 to show the flexibility of the proposed algorithm. Finally, concluding remarks are made in Section 7.

2. Governing equations

2.1. Fluid motion

The dynamics of a two-dimensional, incompressible fluid flow at constant kinematic viscosity ν in a domain \mathcal{D} bounded by $\partial\mathcal{D} \equiv \mathcal{B}$ is governed by the vorticity transport equation

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega, \quad (1)$$

where \mathbf{u} is the velocity, $\omega = \nabla \times \mathbf{u} = \omega \mathbf{e}_z$ the fluid vorticity, and \mathbf{e}_z is a unit vector perpendicular to the plane of the velocity field. The incompressibility condition $\nabla \cdot \mathbf{u} = 0$ allows the definition of a solenoidal streamfunction Ψ , such that $\mathbf{u} = \nabla \times (\Psi \mathbf{e}_z) + \mathbf{U}_\infty$, where \mathbf{U}_∞ is the free-space velocity such that

$$\mathbf{u}(\mathbf{x}) \rightarrow \mathbf{U}_\infty \quad \text{as } |\mathbf{x}| \rightarrow \infty, \quad (2)$$

and the vorticity and the stream function are related through the Poisson equation

$$\nabla^2 \Psi = -\omega. \quad (3)$$

Alternatively, Eq. (3) can be written in integral form by observing that $-\frac{1}{2\pi} \log |\mathbf{x}|$ is the two-dimensional free-space Green's function to ∇^2 . Thus

$$\Psi(\mathbf{x}) = \Psi_\infty + \frac{1}{2\pi} \int \int_{\mathcal{D}} \log |\mathbf{x} - \mathbf{y}| \omega(\mathbf{y}) d\mathbf{y}, \quad (4)$$

and the fluid velocity is

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}_\infty - \frac{1}{2\pi} \int \int_{\mathcal{D}} \frac{(\mathbf{x} - \mathbf{y}) \times \omega(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^2} d\mathbf{y}. \quad (5)$$

2.2. Boundary conditions

The kinematic velocity boundary condition at the solid surface states that the surface is impermeable

$$\mathbf{u}(\mathbf{x}_{\mathcal{B}}) \cdot \mathbf{n}_{\mathcal{B}} = 0, \quad (6)$$

and $\mathbf{n}_{\mathcal{B}}$ is the surface unit normal vector. The velocity boundary condition (6) is related to a boundary condition for the vorticity [32] through the kinematic relation of Eq. (5)

$$\frac{1}{2\pi} \int \int_{\mathcal{D}_{\mathcal{B}}} \frac{(\mathbf{x}_{\mathcal{B}} - \mathbf{y}) \times \omega(\mathbf{y})}{|\mathbf{x}_{\mathcal{B}} - \mathbf{y}|^2} d\mathbf{y} = \mathbf{J}(\mathbf{x}_{\mathcal{B}}) + \mathbf{U}_\infty, \quad (7)$$

where $\mathcal{D}_{\mathcal{B}}$ is a fluid layer of infinitesimal thickness adjacent to the surface \mathcal{B} , and the vector $\mathbf{J}(\mathbf{x}_{\mathcal{B}})$ describes the induced velocity from the vorticity in the fluid excluding the contributions from $\mathcal{D}_{\mathcal{B}}$. Eq. (7) may be written as a surface integral by defining the surface vortex sheet ($\gamma = \gamma \mathbf{e}_z$) through:

$$\frac{\partial \gamma}{\partial n} = \omega, \quad \text{and} \quad \gamma = \lim_{n \rightarrow 0} \int_n \omega dn, \quad (8)$$

where n is the surface normal. Thus

$$\frac{1}{2\pi} \int_{\mathcal{B}} \frac{(\mathbf{x}_{\mathcal{B}} - \mathbf{y}_{\mathcal{B}}) \times \gamma(\mathbf{y}_{\mathcal{B}})}{|\mathbf{x}_{\mathcal{B}} - \mathbf{y}_{\mathcal{B}}|^2} d\mathcal{B} = \mathbf{J}(\mathbf{x}_{\mathcal{B}}) + \mathbf{U}_\infty. \quad (9)$$

The components of the vector equation (9) are Fredholm integral equations in the unknown γ . The two equations (9) are identical to one another cf. e.g. [12], and the solution is unique up to a constant, i.e. an infinite number of solutions exists. The solution is rendered unique by imposing a global constraint on the vorticity, that the time rate of change of the total vorticity be zero (Kelvin's theorem)

$$\frac{d}{dt} \int \int_{\mathcal{D}} \omega d\mathbf{x} = 0. \quad (10)$$

The system of equations (9) and (10) may be solved by removing the singularity of the kernel (Eq. (9)) cf. [33] or, as in the present implementation, by the method of least squares. In most previous works cf. Refs. [34–38] the Eq. (9) is solved for the zero tangential velocity resulting in a Fredholm equation of the second kind. However, the present study follows previous implementations by one of the authors [10] enforcing the normal velocity component, resulting in a Fredholm equation of the first kind. Moreover, the no-slip condition ($\mathbf{u}(\mathbf{x}_{\mathcal{B}}) \cdot \mathbf{s}_{\mathcal{B}} = 0$), where $\mathbf{s}_{\mathcal{B}}$ is the unit tangential vector, is imposed implicitly as a consequence of Eqs. (1) and (9) subject to Eq. (10). It is interesting to note that, assuming the solid is located at $n = 0$ (Eq.

(8)) and the vortex sheet extends from $n = 0$ to $n = 0^+$, both the inviscid and viscous flow can be viewed as satisfying the no-slip condition. The jump in the velocity occurs in the fluid reaching the no-slip velocity condition at the surface at $n = 0$. The inviscid case is a special case of the viscous problem preserving an infinitesimally thin surface vorticity sheet, whereas the viscous flow allows diffusion of the surface vorticity into the flow, thus developing a boundary layer.

2.3. Aerodynamic forces

The total aerodynamic forces are computed from the time derivative of the fluid impulse [32], which yields the total force on the system

$$\mathbf{F} = -\rho \frac{d}{dt} \int \int \omega \times \mathbf{x} d\mathbf{x}. \quad (11)$$

The time derivative is approximated with first order finite differences of the particle impulse.

Pressure forces are computed from the local surface pressure distribution, which can be computed from the local surface flux of vorticity

$$\left. \frac{\partial p}{\partial s} \right|_{\mathcal{B}} = -\rho \left. \frac{\partial \omega}{\partial n} \right|_{\mathcal{B}}. \quad (12)$$

The flux ($\partial \omega / \partial n$) is computed from Eqs. (1) and (8) and neglecting streamwise diffusion, thus

$$\left. \frac{\partial p}{\partial s} \right|_{\mathcal{B}} = -\rho \left. \frac{\partial \gamma}{\partial t} \right|_{\mathcal{B}}. \quad (13)$$

$\partial \gamma / \partial t$ is given by the effective rate of change of the surface vortex sheet cf. [10]. The total pressure forces are finally obtained by integrating the surface pressure

$$\mathbf{F}_p = \oint_{\mathcal{B}} p n d\mathcal{B}. \quad (14)$$

3. Particle vortex methods

In the following, we outline the central components of particle vortex methods for the simulation of flow past bodies of arbitrary shape. The components involve kinematics, convection, boundary conditions imposed using boundary elements, and the modeling of diffusion.

3.1. Kinematics

The classical two-dimensional vortex method is based on the discretization of the vorticity field by a finite sum of N_p Lagrangian particles located at \mathbf{x}_p as follows:

$$\omega(\mathbf{x}, t) = \sum_{p=1}^{N_p} \eta_\sigma(|\mathbf{x} - \mathbf{x}_p|) \Gamma_p. \quad (15)$$

$\Gamma_p(t)$ is the strength of the p th particle and η_σ is a smooth approximation to the delta function, thus $\lim_{\sigma \rightarrow 0} \eta_\sigma(|\mathbf{x}|) = \delta|\mathbf{x}|$, and

$$\eta_\sigma(\mathbf{x}) = \frac{1}{\sigma^2} \eta\left(\frac{|\mathbf{x}|}{\sigma}\right), \quad (16)$$

where σ is the core radius and η a smooth cut-off function. In the present study we employ the second order Gaussian cut-off [39]

$$\eta(r) = \frac{1}{2\pi} \exp\left(-\frac{r^2}{2}\right) \quad (17)$$

throughout.

The velocity field in free space is determined from Eq. (5) using the particles (15) as quadrature points in a Biot–Savart relation

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}_\infty - \frac{1}{2\pi} \sum_{p=1}^{N_p} \mathbf{K}_\sigma(\mathbf{x} - \mathbf{x}_p) \times \Gamma_p \mathbf{e}_z, \quad (18)$$

where the velocity kernel $\mathbf{K}(\mathbf{x}) = \mathbf{x}/|\mathbf{x}|^2$ has been convolved with the cut-off function to obtain $\mathbf{K}_\sigma(\mathbf{x}) = \mathbf{K} * \eta_\sigma(\mathbf{x})$.

3.2. Convection

The vorticity transport equation (1) is solved using an operator splitting technique. The first step involves the solution of the inviscid Euler equation

$$\frac{D\omega}{Dt} = 0, \quad (19)$$

which is approximated by the convection of the vortex elements, thus

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p(\mathbf{x}_p), \quad (20)$$

$$\frac{d\Gamma}{dt} = 0, \quad (21)$$

where the particle velocity is computed from (18) using efficient fast multipole methods [16] or VIC techniques [40–42]. Eq. (20) is solved using standard Runge–Kutta techniques. After this inviscid step, the strength or the position of the particles is furthermore modified to account for the viscous diffusion as described in Section 3.4.

3.3. Boundary element method

The kinematic boundary condition (9) is enforced using a boundary element technique [43,44]. The geometry is discretized assuming piecewise linear panels (\mathbf{x}_i , $i = 1, \dots, N_I$) of approximately uniform length ($\Delta s_i = |\mathbf{x}_{i+1} - \mathbf{x}_i|$) cf. Fig. 1, and a linear variation of the surface vortex sheets (γ_i) rendering the method second order accurate. Enforcing the boundary condition at the panel center point, the discrete approximation of the system of Eqs. (9) and (10) reads

$$\mathbf{M}\boldsymbol{\gamma} = \mathbf{b}, \quad (22)$$

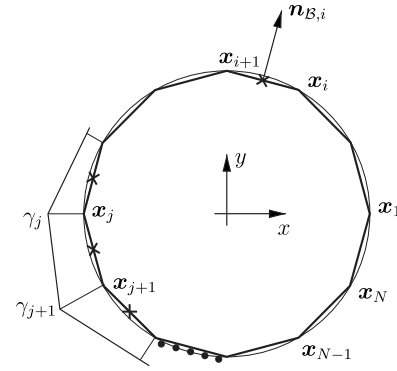


Fig. 1. Schematic of the boundary element discretization. The surface vortex sheet strength is linear along a panel. The boundary condition is enforced at the panel center point (x). Also indicated (•) are the blobs created from the surface sheet strength and to be released in the next step.

where \mathbf{M} is the influence matrix describing the mutual induction of the vortex sheets, and includes a separate equation for the enforcement of the Kelvin theorem. The right-hand side (\mathbf{b}) contains all external influences, and the system of equations is solved in the least squares sense. The influence matrix is constant for non-deforming bodies undergoing an *en bloc* solid body motion and is inverted only once and stored for efficient solution of Eq. (22).

To retain the accuracy and discrete satisfaction of the boundary conditions, the velocity induced by the surface vortex sheets is included in the convection process explicitly as an extra term \mathbf{u}_B to the Biot–Savart equation (18) reading

$$\mathbf{u}_B(\mathbf{x}) = \frac{1}{2\pi} \int_B \frac{(\mathbf{x} - \mathbf{y}_B) \times \boldsymbol{\gamma}(\mathbf{y}_B)}{|\mathbf{x} - \mathbf{y}_B|^2} d\mathcal{B}. \quad (23)$$

At large distances ($|\mathbf{x} - \mathbf{y}_B|/\Delta s > 15$) Eq. (23) is approximated by a point representation cf. Eq. (18), where $\Gamma_i = \gamma_i \Delta s_i$.

3.4. Diffusion modelling

The second step of the fractional step algorithm involves the solution of the diffusion problem

$$\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega, \quad (24)$$

$$\frac{\partial \omega}{\partial n} = 0, \quad \text{on } \mathcal{B}, \quad (25)$$

which is solved using the random walk technique proposed by Chorin [30], see also Refs. [45–48]. The method relies on the probabilistic interpretation of the Green's function solution to Eq. (24) and on the relationship between diffusion and random walk of vorticity-carrying particles. The particle positions resulting from the convection step ($\tilde{\mathbf{x}}_p$) are subjected to a modification as follows:

$$\mathbf{x}_p^{n+1} = \tilde{\mathbf{x}}_p^{n+1} + \boldsymbol{\xi}_p, \quad (26)$$

where ξ_p are random paths according to a Gaussian probability distribution with zero mean and variance $2\nu\Delta t$, and Δt is the time step. This procedure is also adopted for the vorticity release process at the solid boundary where the surface vortex sheet (γ_i) is converted into vortex blobs ($\Gamma_i = \gamma_i \Delta s_i$) and diffused through a random walk into the flow [30]. In order to control the number of particles released in a time step and thus the spatial resolution, a variable number of particles $N_{s_i} = \text{INT}(1 + (N_s^{\max} - 1) |\gamma_i \Delta s_i / \Gamma^{\max}|)$ is created at each panel. The number is chosen according to the strength of the vortex sheet with an upper limited N_s^{\max} corresponding to the panel with the maximum circulation Γ^{\max} . In this study a value of $N_s^{\max} = 6$ was found to provide sufficient resolution of the boundary layer at a tractable computational cost.

Particles entering the solid during a time step may be reflected [49] to satisfy Eq. (25), detained until the next time step as proposed by Smith and Stansby [50] or, as in the present study, deleted as proposed by Chorin [30]. The error associated with the latter approach is equivalent to the approximations of the remeshing considered in Ref. [12]. The remeshing was performed ignoring the presence of the solid boundaries and particles created inside the solids were deleted. The subsequent enforcement of the boundary condition through (22) secured very similar results to their elaborate one-sided remeshing formulae.

Besides being simple to implement, the random walk method enjoys a high robustness, but suffers from a low rate of convergence of $\mathcal{O}(\sqrt{\nu/N_p})$ cf. [51]. Thus, the random walk technique is mostly suited for studies of slightly viscous flows, with small values of ν and with a large number of particles (N_p) required to resolve the details of the flow. Most deterministic diffusion models, on the other hand, require uniformly spaced computational elements to ensure accuracy and stability. This also applies to the PSE scheme [29] which strictly necessitates an occasional re-initialization of the elements. Other models such as that by Shankar and van Dommelen [52] maintain the accuracy between irregularly spaced particles but are computationally expensive. The remeshing required for the convergence of these diffusion models uses one-sided interpolation formulae in the vicinity of solid surfaces [7] with increased algorithmic complexity at immersed interfaces [12]. In the present work we propose a novel partial remeshing strategy which is outlined in Section 5.

4. A novel hybrid particle–mesh algorithm

The computational efficiency of vortex particle methods is closely related to the solution of the N -body problem implied by Eqs. (20) and (18). While the computational cost of the direct method scales as $\mathcal{O}(N_p^2)$, the fast multipole method [16] using adaptive tree data structures [53] reduces the cost to $\mathcal{O}(N_p)$ cf. Refs. [7,10,12]. However, for problems in simple geometries the hybrid particle–mesh Vortex-In-Cell (VIC) algorithm is often preferred

[54,55,40,42,56] due to the availability of efficient solvers for the Poisson equation (3) on a regular mesh. Extensions to the method include domain decomposition techniques and locally body fitted meshes to allow treatment of more complex problems such as the flow past multiple circular cylinders [4,13].

An alternative approach that avoids the generation of curvilinear meshes is the immersed interface method by Peskin [19]. Here the boundary conditions are enforced on the immersed interfaces through momentum source terms acting on the regular mesh. The present work combines the immersed interface method of Peskin with the VIC algorithm using a boundary element description of the interface. While the VIC method generally provides an accurate solution to Eq. (3) for smooth vorticity fields, the discontinuity introduced at an immersed interface results in sub-grid scales not resolved by the VIC method.

Herein, the Particle–Particle Particle–Mesh (P³M) algorithm by Hockney and Eastwood [57] is employed to resolve these sub-grid scales using an influence matrix technique [58], which furthermore allows a disjointed resolution of the mesh and particles. In the following, the classical VIC algorithm is outlined along with the required extensions to arrive at the P³M algorithm.

4.1. The Vortex-In-Cell algorithm

The VIC algorithm is based on a smooth projection of the vorticity carried by the particles onto a regular mesh and the solution of the Poisson equation (3) using fast iterative solvers or direct solvers through Fast Fourier Transforms (FFTs). To this end, particle–mesh techniques approximate the vorticity of (15) by a smooth projection of the particle strengths Γ_p to the grid. For a Cartesian grid consisting of $N_m = N_x \times N_y$ nodes spaced at (h_x, h_y) the relationship corresponding to Eq. (15) for pure particle methods then reads

$$\omega_m(\mathbf{x}_m, t) = \frac{1}{h_x h_y} \sum_{p=1}^{N_p} W(\mathbf{x}_p - \mathbf{x}_m) \Gamma_p. \quad (27)$$

Herein, mesh and particle quantities are denoted by m and p , respectively, and $W(\mathbf{x})$ is the interpolation kernel. Note, that for free-space problems the grid is chosen and possibly adjusted such as to cover the particle map. The projection of the vorticity onto a regular mesh calls for high-order interpolation formula [18]. In this work we implement the third order M'_4 kernel proposed by Monaghan [59] for one-dimensional smooth particle hydrodynamics and later applied for two- and three-dimensional vortex methods [18,42,56]. The kernel reading

$$M'_4(x) = \begin{cases} 1 - \frac{5}{2}|x|^2 + \frac{3}{2}|x|^3, & |x| \leq 1, \\ \frac{1}{2}(2 - |x|)^2(1 - |x|), & 1 < |x| < 2, \\ 0, & |x| \geq 2 \end{cases} \quad (28)$$

conserves the total, linear and angular impulse of the field. Conservation of these first three statistical moments is required to avoid the production of spurious fluid forces [32]. The multidimensional interpolation kernel is constructed as the Cartesian tensor product of the one-dimensional kernel

$$W(x, y) = W(x)W(y). \quad (29)$$

After the projection step (27) the Poisson equation is usually solved by employing Fast Fourier Transforms or fast iterative solvers. For the purpose of this study Fast Fourier Transforms are applied, making use of a technique proposed by Hockney [60] to account for free-space boundary conditions. The procedure involves an extension of the computational domain in the free-space direction(s), cf. Fig. 2, and a mirroring of the Green's function G in the virtual quadrants to exactly cancel the periodic images caused by the FFTs. A one-dimensional schematic of the procedure is shown in Fig. 3.

The velocity field can subsequently be computed from the streamfunction as

$$\mathbf{u} = \nabla \times \Psi \mathbf{e}_z \quad (30)$$

on the mesh using finite differences or through direct differentiation in Fourier space. To this end, the velocity is expressed as a convolution of \mathbf{K} with the vorticity (Eq. (5))

$$\mathbf{u} = \mathbf{K} * \omega, \quad (31)$$

where $\mathbf{K} = \nabla \times (G \mathbf{e}_z)$ is the spatial derivative of the Green's function to the Poisson equation (3), and thus in 2D

$$\mathbf{K}(\mathbf{x}) = \frac{1}{2\pi} \frac{\mathbf{e}_z \times \mathbf{x}}{|\mathbf{x}|^2}. \quad (32)$$

The free-space solution ($\bar{\mathbf{K}}$) for the velocity is then obtained by the following mirroring for the two components of $\mathbf{K} = (K^x, K^y)^T$

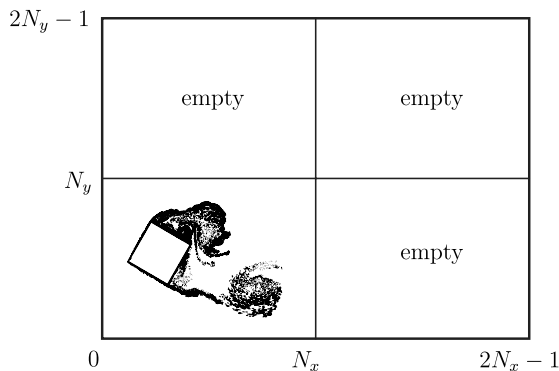


Fig. 2. Schematic of the free-space boundary treatment for use with FFTs.

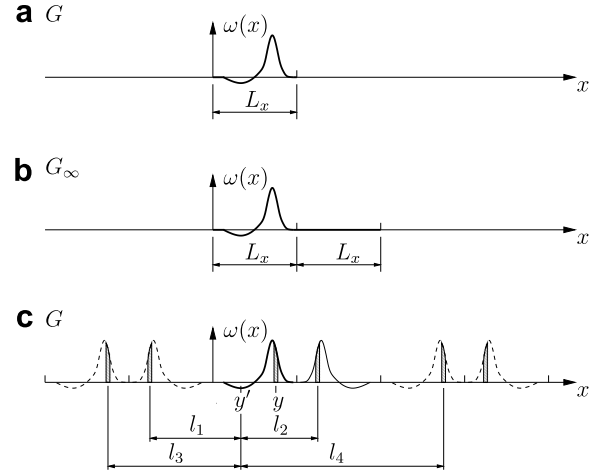


Fig. 3. The free-space solution of the Poisson equation (3) is obtained using Fast Fourier Transforms in an extended domain as proposed by Hockney [60]. The vorticity in the extended region (b) is initialized to zero and the extended one-dimensional Green's function: $G_\infty(k) = G(k)$ for $k < N$, $G_\infty(k) = G(N - k)$ for $N < k < 2N$, and $G(0) = 1$ yields the exact free-space solution. The procedure corresponds to a convolution in a real space over the pairs of periodic images (c). The contribution to the convolution from the images ($l_1 = y + y'$, $l_2 = 2L_x - y - y'$, $l_3 = 2L_x - y + y'$, $l_4 = 2L_x + y - y'$, etc.) is $\sum_k k 2L_x \int \omega dx = 0$ provided the total vorticity is zero.

$$\bar{K}_{i,j}^x = +K_{i,j}^x \left\{ \begin{array}{l} 0 \leq i < N_x \\ 0 \leq j < N_y \end{array} \right\} \quad (33)$$

$$\bar{K}_{2N_x-1-i,j}^x = +K_{i+1,j}^x \left\{ \begin{array}{l} 0 \leq i < N_x - 1 \\ 0 \leq j < N_y \end{array} \right\} \quad (34)$$

$$\bar{K}_{i,2N_y-1-j}^x = -K_{i,j+1}^x \left\{ \begin{array}{l} 0 \leq i < N_x \\ 0 \leq j < N_y - 1 \end{array} \right\} \quad (35)$$

$$\bar{K}_{2N_x-1-i,2N_y-1-j}^x = -K_{i+1,j+1}^x \left\{ \begin{array}{l} 0 \leq i < N_x - 1 \\ 0 \leq j < N_y - 1 \end{array} \right\} \quad (36)$$

$$\bar{K}_{i,N_y}^x = \bar{K}_{N_x,j}^x = \bar{K}_{i,N_y}^y = \bar{K}_{N_x,j}^y = 0 \left\{ \begin{array}{l} 0 \leq i < 2N_x \\ 0 \leq j < 2N_y \end{array} \right\} \quad (37)$$

The convolution (31) is then performed in Fourier space as

$$\hat{\mathbf{u}} = \hat{\mathbf{K}} \hat{\omega}. \quad (38)$$

The resulting velocity field \mathbf{u} in the real quadrant is the solution to the free-space problem. If this exact differentiation is used in the Vortex-In-Cell method, the only source of errors remaining is the particle-mesh interpolation.

The resolved velocity field is finally interpolated back onto the particles using

$$\mathbf{u}_p = \sum_{m=1}^{N_m} W(\mathbf{x}_p - \mathbf{x}_m) \mathbf{u}_m, \quad (39)$$

to allow integration of Eq. (20).

Let us consider the velocity field induced by a single vortex blob to study the accuracy of the PM scheme. In Fig. 4 the mesh-resolved velocity field along a line in the x -direction is shown for a particle located at different positions with respect to the grid. The discrepancy to the

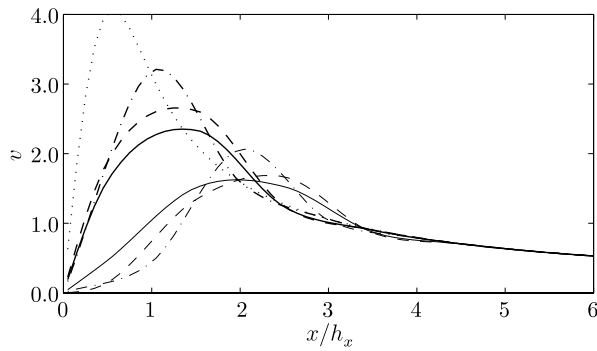


Fig. 4. The particle-mesh resolved free-space velocity induced by a single particle located at different positions and measured along the x -direction. The (\cdots) exact solution using a Gaussian core of radius $\sigma = \frac{1}{2}h_x$ is compared to PM solutions with particles located: $(—)$ at the centre of a cell at $[0, 0]$, $(- -)$ at $[0, \frac{1}{2}h_x]$, $(- \cdot -)$ at $[\frac{1}{2}h_x, \frac{1}{2}h_x]$; (thin lines) are results by fourth order finite differencing of the exact streamfunction solution, (bold lines) are results using the exact velocity computation on the mesh. All results are obtained using the M'_4 interpolation kernel.

direct interaction and the anisotropy of the PM scheme is apparent from this. These inaccuracies stem from unresolved sub-grid scales which arise from both the mesh interpolation and the finite differencing. Fig. 5 illustrates the superiority of the direct velocity computation to the finite differences.

4.2. The Particle–Particle Particle–Mesh algorithm

The Particle–Mesh algorithm described in Section 4.1 requires that Eq. (27) is a close approximation to Eq. (15). This is generally true for systems where either the mean particle spacing is larger than the mesh spacing ($h/h_x > 4$ cf. Fig. 4) or for systems with smooth vorticity fields. Flows around immersed interfaces clearly exhibit discontinuities in the vorticity field such that the particle discretized velocity field \mathbf{u}_p contains sub-grid scales \mathbf{u}_{pp} with

$$\mathbf{u}_p = \tilde{\mathbf{u}}_p + \mathbf{u}_{pp}, \quad (40)$$

where $\tilde{\mathbf{u}}_p$ is the grid-resolved velocity.

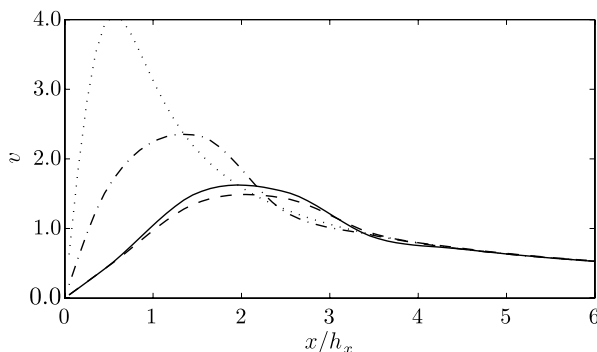


Fig. 5. The particle-mesh resolved free-space velocity induced by a single particle located at the centre of a cell at $[0, 0]$ and measured along the x -direction. The (\cdots) exact solution using a Gaussian core of radius $\sigma = \frac{1}{2}h_x$ is compared to: $(- -)$ second order finite differences, $(- \cdot -)$ fourth order finite differences, $(—)$ exact velocity computation on the mesh. The M'_4 kernel is used for the PM projection.

The accuracy of the direct Particle–Particle (PP) interaction and the efficiency of the Particle–Mesh (PM) algorithm have been successfully combined in the hybrid Particle–Particle Particle–Mesh (P^3M) algorithm first proposed by Hockney and Eastwood [57].

The present study makes use of an influence matrix algorithm proposed by Walther [61]. It is similar to the methods of Anderson [62], Theuns [63], and Phillips et al. [64] but provides an exact computation of the mesh-resolved field by an influence matrix technique. This matrix represents the approximations and anisotropy caused by the differential operators on the mesh. During the projection step the resolved velocity field induced by a number of particles contained within a cell is computed as

$$\delta \mathbf{u}_i = \mathbf{C} \delta \boldsymbol{\omega}_j, \quad (41)$$

where $\delta \mathbf{u}_i$ is the velocity at node i induced by a nodal vorticity $\delta \boldsymbol{\omega}_j$ at node j . This influence is localized over L nodes, the near-range region of each cell, cf. Fig. 6.

The influence matrix then has dimension $L \times M$, where M is the kernel's number of supporting nodes and the dimension of vector $\delta \boldsymbol{\omega}_j$. For the two-dimensional M'_4 kernel and N_r neighboring cells included in the PP correction $M = 4 \times 4$ and $L = (2N_r + 2)^2$, cf. Fig. 6. The nodal velocity values computed by Eq. (41) are subsequently subtracted on the respective nodes, relying on the linearity of the Poisson equation to cancel exactly the PM contribution of the particles contained in the originating cell. In the following PP step the exact $|\mathbf{x}|^{-1}$ velocity correction (Eq. (18)) between the particles within the neighborhood provides the final velocity prediction of the P^3M algorithm.

The influence matrix is constructed by placing M individual test particles of unit strength at arbitrary positions in a central cell of the domain and evaluating the corresponding PM-predicted velocity field. The matrix is insen-

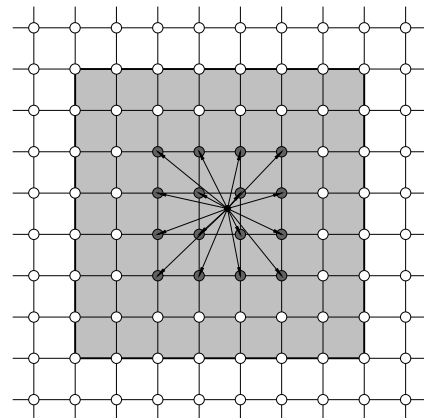


Fig. 6. Schematic of the P^3M algorithm for the M'_4 interpolation kernel and three neighbouring cells ($N_r = 3$) included in the particle–particle correction. The particle (\bullet) assigns to the $M = 16$ surrounding nodes. Indicated by the shaded area are the $L = 64$ nodes included in the influence matrix prediction of the velocity field.

sitive to the locations of the test particles but they should be placed sufficiently far apart to avoid a singular matrix. The projected vorticity of the k th test particle on the m th grid point $\delta\Omega_{m,k}$ is determined using (27). Next, the velocity field is computed using the Poisson solver employed in the PM part of the algorithm and sampled at the L nodes neighboring the sample cell as $\delta U_{i,k}$. Assembling this for the M test particles forms the following linear system of equations:

$$\delta U_{i,k} = C_{i,m} \delta\Omega_{m,k} \quad (42)$$

which can be solved for the elements of C .

Summarizing, the influence matrix P³M technique proceeds as follows:

- (1) Project the particle strengths Γ_p to the mesh using Eq. (27).
- (2) Solve the Poisson equation $\nabla^2 \Psi = -\omega$ on the mesh.
- (3) Compute the velocity field $\mathbf{u} = \nabla \times \Psi \mathbf{e}_z$ on the mesh. (Steps 2 and 3 can be combined if the velocity is computed directly as proposed in Section 4.1.)
- (4) Interpolate the resolved velocity field back onto the particles using Eq. (39), which yields $\tilde{\mathbf{u}}_p$.
- (5) For each non-empty grid cell perform the following steps to estimate the PM-resolved velocities at the neighboring cells induced by the projected vorticity $\delta\omega_m$:
 - (a) Compute the velocity at the neighboring grid nodes by applying the influence matrix (Eq. (41)): $\delta\mathbf{u}_m = C\delta\omega_m$.
 - (b) For each of the neighbouring cells project the resulting velocity to the particles to obtain \mathbf{u}_{ps} .
- (6) Compute the sub-grid velocities as a local particle–particle interaction correction \mathbf{u}_{pc} using Eq. (18).
- (7) Contributions from external velocity potentials are included as \mathbf{u}_{pe} .
- (8) The total particle velocity is $\mathbf{u}_p = \tilde{\mathbf{u}}_p - \mathbf{u}_{ps} + \mathbf{u}_{pc} + \mathbf{u}_{pe}$.

Let us consider again the studies on a single particle from Section 4.1 which illustrated the unresolved sub-grid scales from the PM method. The velocity resulting from the P³M scheme is shown in Fig. 7. The grid used for the P³M analysis is also indicated, with the particle located in the middle of a cell. Results of the P³M algorithm for two different cut-off ranges N_r are provided. Clearly, within the direct-interaction range the exact solution is obtained. Errors arise outside of the PP-interaction cut-off due to the finite-differences employed here. The results indicate that, in order to keep this error below 1%, $N_r = 4$ neighbors should be included in the correction. If the velocity is computed using exact differentiation in Fourier space, $N_r = 3$ is sufficient (not shown).

4.3. Implementation issues

In this section we discuss a few issues regarding a flexible and efficient implementation of the presented algorithm.

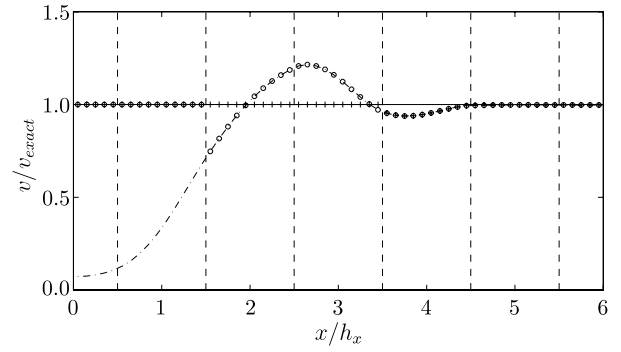


Fig. 7. Induced velocity from a single particle in a 2D domain relative to the exact solution using a Gaussian core of radius $\sigma = \frac{1}{2}h_x$. Dashed lines indicate the cell boundaries. Results are obtained using (—) direct PP interaction; (---) PM (M_4' kernel, 4th order finite differences); marked lines with P³M of different cut-off: (—o—) ($N_r = 1$), (—+—) ($N_r = 3$).

The method leaves complete freedom regarding the treatment of immersed boundaries and their surface vorticity. As discussed in Section 3.3 we employ a linear discretization of the vortex sheet strength. This is incorporated in the P³M algorithm by replacing each linear element by a point vortex of the corresponding strength $\Gamma_i = \frac{1}{2}(\gamma_j + \gamma_{j+1})\Delta s_j$. These point vortices then form an approximation of the sheets within the PM and influence matrix assignment part. This ensures a sufficient representation of the surface circulation in regions far away from the surface, whereas the effect of the surface point vortices on near-field particles is cancelled out and replaced by the near-field sub-grid correction step. Any surface modelling can then be applied in the PP step of the algorithm. Whilst herein a linear vortex sheet strength approximation is used, any other approach would be equally possible.

With the influence matrix technique being based on the vorticity projected from the particles to the grid nodes, considerable time-savings can be accomplished by saving the mesh-interpolated vorticity contribution of each cell separately. This procedure requires the storage of M vorticity values per node to distinguish the origin of the contributions. Subsequently, the respective elements of this matrix can be accessed to construct the $\delta\omega_m$ vector of Eq. (41). Furthermore, the computed nodal velocity components of the local influence-matrix prediction are stored and subsequently subtracted from the PM solution before a final interpolation back to the particles. Thus, only one forward and one backward interpolation is needed per particle.

Since the anisotropy of the flow fields is usually largest close to the solid surface, this calls for a variable N_r if a uniform accuracy throughout the domain is sought. If one is prepared to give up the use of symmetry for some interactions within the PP part, the maximum value of N_r (cf. Section 4.2) would be used for cells in the proximity of the wall only and then successively reduced according to some function of the distance of the cell from the solid boundary. This significantly speeds up the procedure and is particularly suited for the simulation of the wake flow.

The P³M algorithm is furthermore used in the construction of the right-hand side of the boundary element equation. Eq. (22) requires the calculation of the residual velocity on the surface induced by the free vortices. In the naïve implementation this operation is of $\mathcal{O}(N_p N_t)$. However, to accomplish this efficiently, ghost particles are introduced at the sampling points (panel centers). These are no numerical elements but rather dummy particles representing required target points for the velocity computation. Since the P³M algorithm can handle disjoint source and target points, a call to the P³M routine by passing the ghost particles on will result in a strongly reduced computation, because computations related to the velocity target points (back-projection from mesh to particle and particle–particle correction) are only performed for the ghost particle locations.

It is important to note that, while herein we utilize Fast Fourier Transforms for the solution of the Poisson equation, the proposed P³M algorithm does not require access to Fourier space contrary to the classical P³M algorithm. Therefore alternatively “black-box” fast field solvers can be used, which may be important for efficient parallelization [65].

The current algorithm can be readily extended to account for movements of the solid interface such as to perform simulations of Fluid-Structure Interaction (FSI) problems [66]. Due to the disjoint nature of interface and computational grid the solid surface can be allowed to displace without the need for adjustments to the grid. The current implementation has been used extensively to study FSI problems, mainly in the field of bluff body aerodynamics such as bridge aerodynamics [67,68].

5. Particle remeshing

Herein we introduce a novel procedure for the re-initialization of the particle locations. Such remeshing is necessary for the convergence of the vortex method as discussed in Section 3.4. It proceeds by replacing the irregularly spaced vortex elements by a new set of particles located on a regularized lattice – the particle mesh. To this end, the particle strengths Γ_p are interpolated to the nodes of the particle mesh using the particle–mesh interpolation formula (27). These mesh nodes then form a new regularized set of particles replacing the old. The moment conserving properties of the interpolation used in the remeshing step are essential for preserving the dynamics of the flow. Thus, we employ the third order M_4' (Eq. (28)) throughout.

Difficulties arise close to solid boundaries which require not to assign vorticity inside the body. For diffusion models which strictly require a regular particle layout, one-sided kernels are usually used to perform the remeshing in the vicinity of the surface. The random walk model used herein however is less sensitive to the particle map as discussed earlier. This robustness allows for a scheme where we do not replace particles which would assign vorticity inside the body. This yields a thin region along the surface in

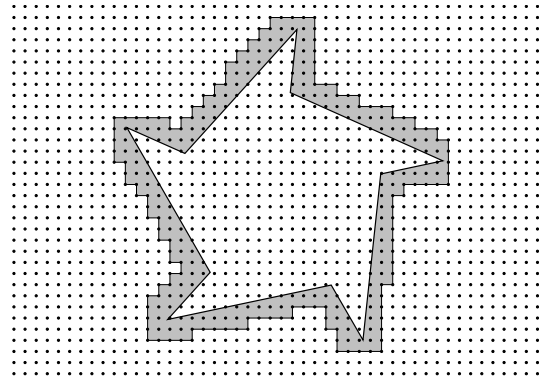


Fig. 8. Schematic of the partial remeshing strategy: dots indicate the particle mesh; particles in the shaded area are not being remeshed.

which the distorted vortex layout is conserved as depicted in Fig. 8. Since this region usually contains a large number of particles due to the vortex release algorithm, the overlap of particle cores is ensured without remeshing. In the remaining flow regions the redistribution ensures the overlap and thus a sufficient vorticity support.

For engineering applications, a method has also been devised to take into account the reduced need for resolution in the far-field regions. There, the particle spacing can be increased without significant loss of accuracy. Hence, a zonal remeshing is used where the particle mesh is successively coarsened away from the surface as shown exemplary in Fig. 9 for the flow past a bridge deck [67]. The different particle meshes are laid out in zones Ω_i and can be placed independent as far as node location and spacing are concerned. By definition a particle is remeshed only to nodes of one particle mesh, applying this mesh's interpolation rule. Two adjacent zones Ω_i and Ω_{i+1} will have a certain overlap, the size of which depends on the interpolation kernels. This overlap of the resulting particle maps also ensures sufficient vorticity support without gaps between the zones. Note, that the core radius must be altered according to the different particle mesh spacings, cf. also [69]. Interactions between vortex blobs with different radii are based on the average radius. The zonal remeshing reduces the number of particles and proves useful in adjusting the computational effort to the required numerical resolution.

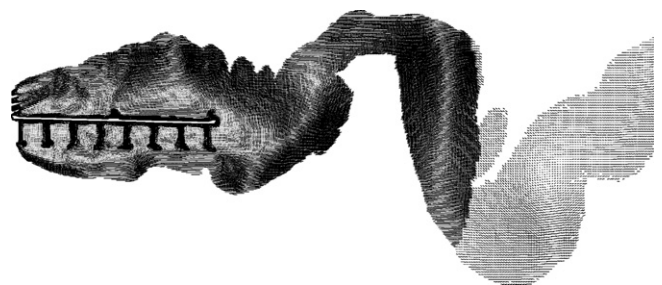


Fig. 9. Sample particle map for a bridge deck study using zonal remeshing.

6. Results

In the following we demonstrate the convergence of the proposed method by considering simple inviscid, steady flow, and the impulsively started flow past a circular cylinder. To illustrate the ability of the method to handle complex geometries we present a study of the impulsively started flow past a cactus-like geometry.

6.1. Circular patch of vorticity

To study the convergence of the proposed P³M algorithm we consider the velocity field induced by a circular patch of vorticity under free-space boundary conditions. To this end, the vorticity patch introduced by Perlman [70] is used, thus the initial vorticity field is given by

$$\omega(\mathbf{x}) = \begin{cases} (1 - |\mathbf{x}|^2)^7 & |\mathbf{x}| \leq 1, \\ 0 & |\mathbf{x}| > 1. \end{cases} \quad (43)$$

For the PM component of the algorithm the third order M₄' interpolation kernel and fourth order finite-differences are used. The particle–particle correction term and the reference N_p^2 solution both employ the Gaussian kernel Eq. (17).

Assuming an inviscid flow, the vorticity distribution of Eq. (43) results in a stationary flow. Thus, in this study we only consider the spatial convergence of the algorithm and compare the computed velocity field ($\mathbf{v}_p^{\text{P}^3\text{M}}$) with the corresponding N_p^2 solution ($\mathbf{v}_p^{\text{dir}}$). No time stepping or remeshing is performed. Different computational meshes are employed, each covering a domain of $[-1.2; 1.2]$ in both directions. The core radius is kept constant at $\sigma = 0.005$ throughout. The number of particles is kept constant at four per grid cell. The particle field is initialized by placing particles on a separate grid in order to discretize the vorticity distribution of Eq. (43).

Fig. 10 shows the convergence of the L_2 error of the particle velocity

$$L_2(v) = \frac{1}{N_p} \sqrt{\sum_{p=1}^{N_p} |\mathbf{v}_p^{\text{dir}} - \mathbf{v}_p^{\text{P}^3\text{M}}|^2} \quad (44)$$

obtained using the hybrid PM and P³M algorithms. The PM solution displays a third order convergence as expected from a combination of third order interpolation and fourth order finite differences. The simulations with the P³M algorithm were obtained by increasing the number of neighboring cells as the mesh spacing decreases, thus keeping the geometric size of the direct-interaction region constant. The error is reduced with respect to the PM solution as a result of the local particle–particle correction. The order of convergence is approximately 5 for the case studied here, but depends on the selection of N_r .

The overall convergence rate of the current algorithm will be subject to the combined effects of the different sub-algorithms and due to the use of random walks lower than that of the velocity computation shown above. How-

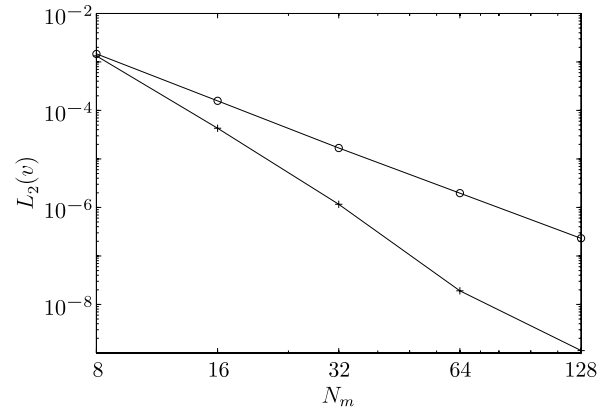


Fig. 10. Convergence of the L_2 -error of the particle velocities as compared to the direct particle–particle interaction computation. The number of particles per grid cell is kept constant. (—○—) PM prediction (M₄' kernel, 4th order finite differences); (—*—) P³M with N_r such that the PP-correction area remains constant.

ever, it is important for the overall performance to keep the individual errors low. The P³M is thus of great importance.

6.2. Flow past an impulsively started circular cylinder at $Re = 3000$

In this section validation studies for the case of an impulsively started flow past a circular cylinder are presented. This problem is a standard benchmark in computational fluid dynamics. Thus, extensive high-resolution studies have been carried out by Koumoutsakos and Leonard [7] and Ploumhans and Winckelmans [12] presented several extensions to the 2D vortex method also analyzing this particular flow. For the purpose of this validation study analyses at a Reynolds number $Re = U_\infty D/\nu$ of 3000 are performed. To show convergence we systematically refine the adjustable parameters of the method. Moreover, we investigate the behavior of the proposed partial remeshing strategy.

A reference case is defined by the following parameters. The surface of the cylinder is discretized by $N_l = 400$ panels, from which at most (N_s^{max}) 6 vortex particles are created. The velocity field is computed using the P³M algorithm with a grid resolution of 256×128 and fourth order finite differences. The particle–particle correction step is performed within four neighboring cells ($N_r = 4$). The projection steps all use the third order M₄' kernel. A second-order Runge–Kutta scheme is applied for the time integration of the convection step with a non-dimensional time step of $\Delta t^* = \Delta t U_\infty/\Delta s = 0.5$. A core radius of $\sigma/\Delta s = 1.2$ is used throughout. All results of the convergence study are shown in Fig. 11. Shown are the time history of the drag coefficient filtered using a running average with a window size of 10 time steps.

The reference solution is compared with previously published studies [7,12,13]. In spite of the stochastic nature of the method due to the use of random walks and hence

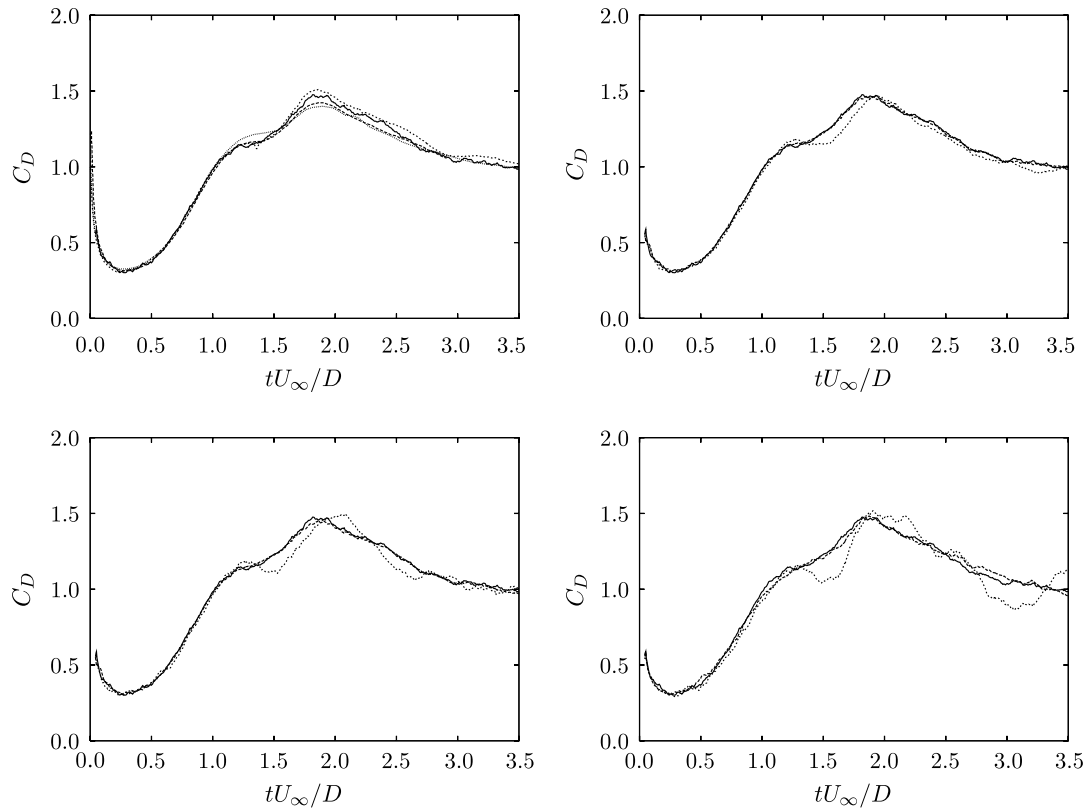


Fig. 11. The spatial convergence of the drag coefficient for the impulsively started flow past a circular cylinder at $Re = 3000$. Top left: Present results, reference case: (—); Previous results: (---) Koumoutsakos and Leonard [7], (---) Ploumhans and Winckelmans [12], (---) Ould-Salihi et al. [13]. Top right: Variation of the number of boundary elements: (—) reference: $N_l = 400$; (---) $N_l = 300$; (---) $N_l = 600$. Bottom left: Variation of the maximum number of particles released from a boundary element: (—) reference: $N_s^{\max} = 6$; (---) $N_s^{\max} = 4$; (---) $N_s^{\max} = 8$. Bottom right: Variation of the remeshing strategy: (—) reference: unremeshed; (---) remeshed $h_p = \Delta s/2$; (---) remeshed $h_p = \Delta s/4$. Present results are plotted after filtering.

formally low convergence rate, the present results are found to be in excellent agreement with these previous studies all employing the deterministic, second order PSE scheme [29] for diffusion.

First we consider the spatial convergence by varying the number of boundary panels (N_l) and the vortex creation

parameter N_s^{\max} . The results obtained using 400 and 600 panels are indistinguishable, whereas the simulation using 300 panels displays some deviation toward the time of maximum drag. Similarly, values of N_s^{\max} of 6 and 8 produce virtually identical results whereas $N_s^{\max} = 4$ results in an insufficient particle resolution.

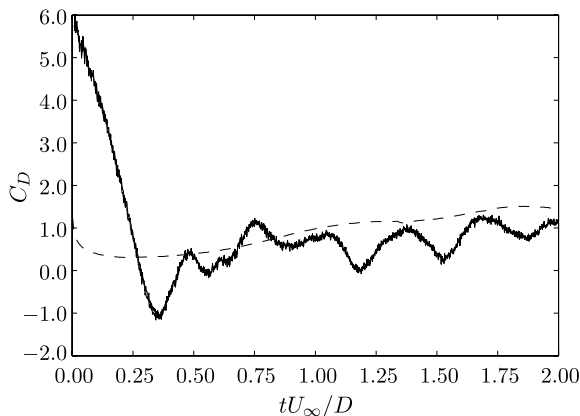


Fig. 12. The time history of the drag coefficient for the impulsively started flow past a 12 leaf cactus at $Re = 3000$: (—) $L/D = 0.250$ (12 leaves), pressure drag; (---) $L/D = 0.250$ (12 leaves), total drag; (---) $L/D = 0.000$ (∞ leaves), total drag.

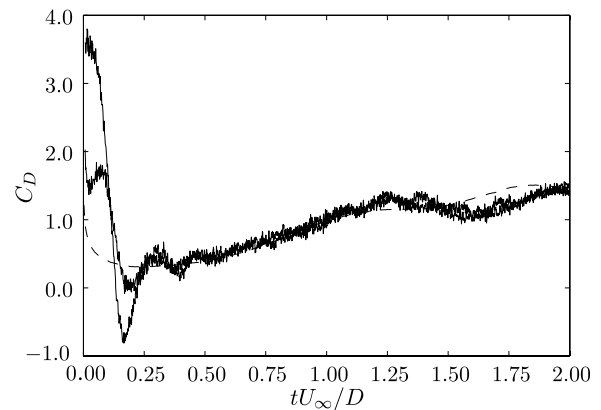


Fig. 13. Impulsively started flow past cacti with different corrugation depths at $Re = 3000$. The time history of the total drag coefficient: (—) $L/D = 0.035$ (24 leaves), (---) $L/D = 0.070$ (24 leaves), (---) $L/D = 0.000$ (∞ leaves).

Next, the partial remeshing strategy introduced in Section 5 was tested using two different mesh resolutions and the results compared with a solution obtained without remeshing. The remeshing is performed at every time step and no suppression of low-strength vortices is used. For this particular flow configuration, and at early times ($tU_\infty/D < 3.5$) the unremeshed simulation appears to have converged and is found to be in good agreement with the partially remeshed solution with a mesh spacing (h_p) of $\Delta s/4$. The solution obtained using double the spacing ($\Delta s/2$) results in sufficient resolution at early times but suffers from deviations at later times ($tU_\infty/D > 1.5$).

6.3. Impulsively started flow past a cactus-like geometry

The ability of cacti and other tall succulents to withstand high wind speeds has recently motivated experimental and numerical studies [71] of the aerodynamic behavior of these structures. The corrugated surface of cavities and spines was found to reduce the mean drag and the fluctuating side-force for a 24 leaf cactus with cavity depths (L/D) as low as 0.035. The arrangement of the cavities and spines of the Saguaro cactus (*Cereus giganteus*) are speculated to have evolved in nature due to advantages in the process of natural selection related to the aerodynamic forcing.

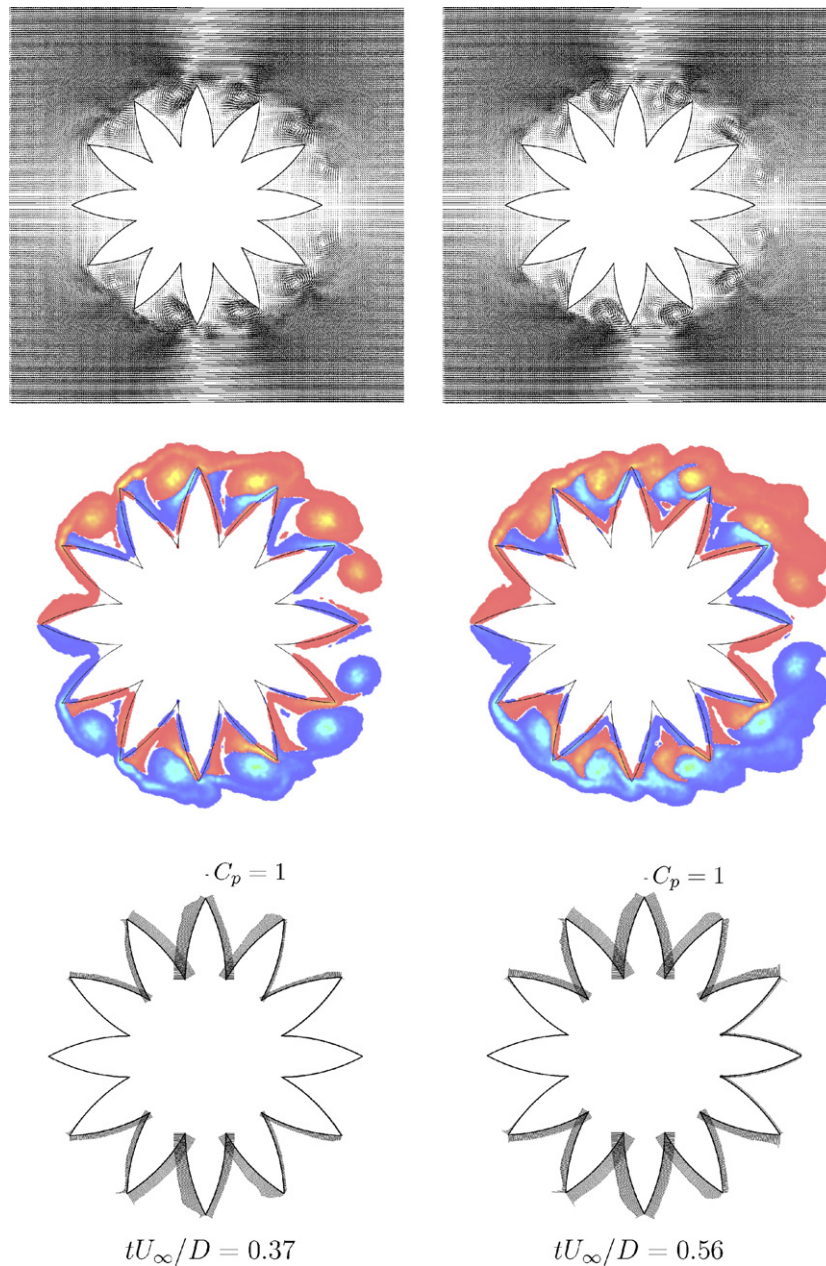


Fig. 14. Impulsively started flow past a 12 leaf cactus with $L/D = 0.250$ at $Re = 3000$. The snapshots show the velocity (top) and vorticity (middle) fields and the surface pressure distribution (bottom) at the occurrence of the first two drag minima at times $tU_\infty/D = 0.37$ (left) and $tU_\infty/D = 0.56$ (right). The resulting drag is negative in both cases. The minima occur as the vortices in the cavities are convected to the downstream side of the cavity resulting in a low local surface pressure.

The present study involves the impulsively started flow past three different surface geometries: two cacti with 24 leaves and cavity depths of 0.035 and 0.070, respectively, and one cactus with 12 leaves and a cavity depth of 0.250. The simulations are performed at a Reynolds number of 3000, and conducted until a non-dimensional time (tU_∞/D) of 2 with a time step of $\Delta tU_\infty/D = 0.0015$. The number of boundary elements is chosen such as to obtain panel lengths of approximately $\Delta s/D = 0.002$, corresponding to 1680, 2160 and 2640 elements for the three cases. Partial remeshing is used at every time step, with a particle mesh resolution of $h_p = \Delta s/2$. The VIC mesh spacing is $h/h_p = 4$, resulting in a mesh size of (1024, 512) at the end of the simulation. The range of local particle–particle correction is $N_r = 4$. This high resolution was found to be necessary in a convergence study which involved a successive refinement of the spatial and temporal parameters. All simulations involved roughly one million particles at completion.

The time history of the drag force obtained for the 12 leaf cactus is shown in Fig. 12. For comparison the drag force on a circular cylinder is also shown. Both the pressure and total drag are displayed revealing a negligible viscous drag term. Fig. 13 shows the total drag force time history for the remaining surface configurations. At early times ($tU_\infty/D < 0.1$) the corrugated cylinders exhibit a marked increase in the drag of 1.8, 3.6 and 5.8 compared to a value of approximately 1.0 for the circular cylinder. This increased initial drag is caused by the formation of the vortices at the tip of the spines. As these vortices detach the drag decreases and reaches a minimum when the vortices touch the adjacent downstream spine. The low surface pressure in the vicinity of the vortices is responsible for the observed negative drag. As the vortices diffuse, or are being convected out of the cavity, the drag recovers to a value in close agreement with that of a circular cylinder. The 12 leaf case however experiences a second drag minimum as the vortices in the upstream pointing cavities touch their downstream spine wall at approximately $tU_\infty/D = 0.56$. Flow visualization and the surface pressure distribution at these minima are shown in Fig. 14. The vorticity field is here computed using Eq. (15) evaluated on a 1000×1000 mesh.

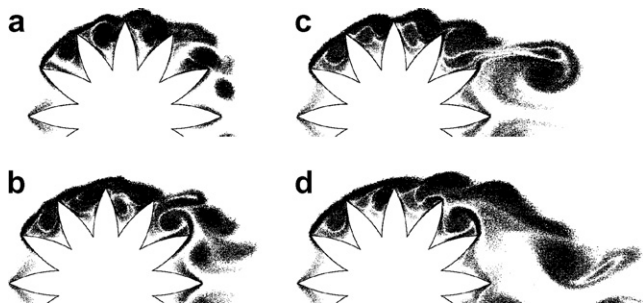


Fig. 15. Impulsively started flow past a 12 leaf cactus with $L/D = 0.250$ ($Re = 3000$), snapshots of the particle map: $tU_\infty/D = 0.5, 1.0, 1.5$ and 2.0 (a–d).

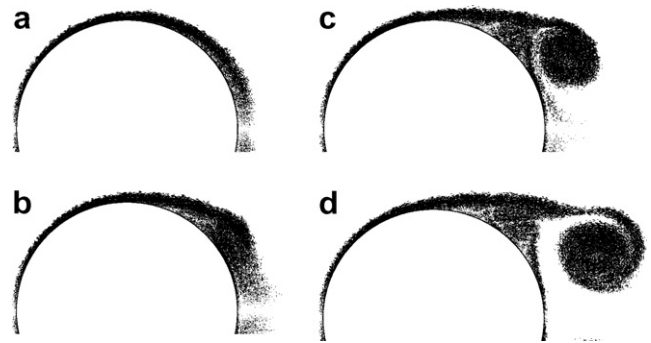


Fig. 16. Impulsively started flow past a circular cylinder with $L/D = 0.000$ ($Re = 3000$), snapshots of the particle map: $tU_\infty/D = 0.5, 1.0, 1.5$ and 2.0 (a–d).

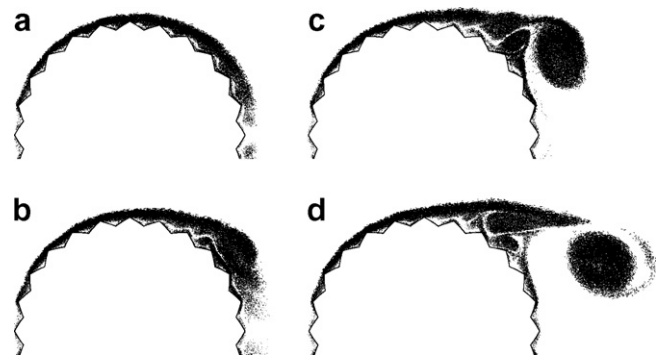


Fig. 17. Impulsively started flow past a 24 leaf cactus with $L/D = 0.035$ ($Re = 3000$), snapshots of the particle map: $tU_\infty/D = 0.5, 1.0, 1.5$ and 2.0 (a–d).

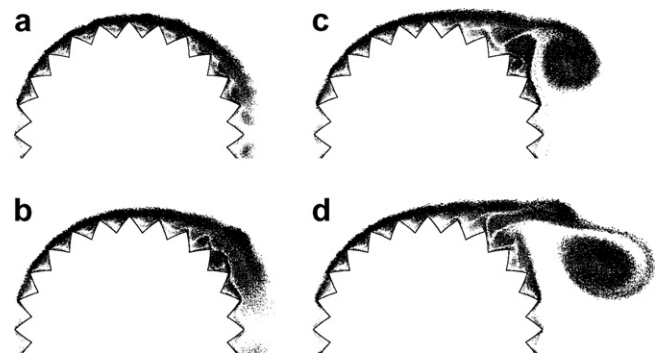


Fig. 18. Impulsively started flow past a 24 leaf cactus with $L/D = 0.070$ ($Re = 3000$), snapshots of the particle map: $tU_\infty/D = 0.5, 1.0, 1.5$ and 2.0 (a–d).

Particle maps for the three different cases at different instances in time are shown in Figs. 15–18. For these plots a threshold on the particle strength has been introduced, resulting in approximately 20% of the particles being shown.

Saguaro reach reproductive maturity after 30–50 years only. Resistance against rare and strong wind gusts is a

natural selection criterion. The present study thus suggests, that their shape has evolved due to the favorable aerodynamic behavior.

7. Conclusions

An immersed interface P³M algorithm has been presented for the simulation of two-dimensional bluff body flows. The Particle–Mesh (PM) step of the algorithm extends the classical Vortex-In-Cell algorithm by employing a novel influence matrix technique to cancel the anisotropic sub-grid scales introduced by the presence of the immersed interface. The subsequent Particle–Particle (PP) step involves an exact particle–particle correction term. The P³M algorithm furthermore allows to disjoin the mesh and particle resolution by explicitly resolving sub-grid scales on the particles.

The random walk technique is used to model diffusion and allows for a partial remeshing strategy in which particles in close proximity of solid boundaries are not remeshed. This leads to a robust diffusion model which ensures the convergence of the method. The choice of random walks however is independent of the P³M velocity computation algorithm.

The simulations of the impulsively started flow past a circular cylinder at a Reynolds number of 3000 demonstrated the convergence of the proposed methodology. The predicted drag time history was found to be in good agreement with previous simulations. The flow past a cactus-like geometry was conducted to demonstrate the ability of the method to handle complex geometries. The simulations revealed a momentary negative drag for cavity depths of $L/D = 0.070$ and 0.250 , respectively.

Acknowledgements

The authors gratefully acknowledge the helpful comments received from Petros Koumoutsakos and Allan McRobie. Guido Morgenthal acknowledges support by Studienstiftung des deutschen Volkes and EPSRC. Jens H. Walther acknowledges partial support by Julie Damms's Studiefond.

References

- [1] Spalart PR. Simulation of rotating stall by the vortex method. *J Propulsion Power* 1985;1(3):235–41.
- [2] Sarpkaya T, Ihrig CJ. Impulsively started steady flow about rectangular prisms: experiments and discrete vortex analysis. *J Fluids Eng* 1986;108:47–54.
- [3] Sethian JA, Ghoniem AF. Validation study of vortex methods. *J Comput Phys* 1988;74:283–317.
- [4] Slaouti A, Stansby PK. Flow around two circular cylinders by the random-vortex method. *J Fluids Struct* 1992;6:614–70.
- [5] Scolan Y-M, Faltinsen OM. Numerical studies of separated flow bodies with sharp corners by the vortex in cell method. *J Fluids Struct* 1994;8:201–30.
- [6] Gharakhani A, Choniem AF. Simulation of the piston driven flow inside a cylinder with an eccentric port. *J Comput Phys* 1998;120(2):319–26.
- [7] Koumoutsakos P, Leonard A. High-resolution simulation of the flow around an impulsively started cylinder using vortex methods. *J Fluid Mech* 1995;296:1–38.
- [8] Koumoutsakos P, Shiels D. Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate. *J Fluid Mech* 1996;328:177–227.
- [9] Lin H, Vezza M, Galbraith RAM. Discrete vortex method for simulating unsteady flow around pitching aerofoils. *AIAA J* 1997;35(3):494–9.
- [10] Walther JH, Larsen A. Discrete vortex method for application to bluff body aerodynamics. *J Wind Eng Ind Aerodyn* 1997;67–68:183–93.
- [11] Larsen A, Walther JH. Aeroelastic analysis of bridge girder sections based on discrete vortex simulations. *J Wind Eng Ind Aerodyn* 1997;67–68:253–65.
- [12] Ploumhans P, Winckelmans GS. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *J Comput Phys* 2000;165:354–406.
- [13] Ould-Salihi ML, Cottet G-H, El Hamraoui M. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM J Sci Comput* 2000;22(5):1655–74.
- [14] Taylor IJ, Vezza M. Application of a discrete vortex method for the analysis of suspension bridge deck sections. *Wind Struct* 2001;4(4):333–52.
- [15] Barnes J, Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* 1986;324(4):446–9.
- [16] Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J Comput Phys* 1987;73:325–48.
- [17] Monaghan JJ. Particle methods for hydrodynamics. *Comput Phys Rep* 1985;3:71–123.
- [18] Cottet G-H, Koumoutsakos P. *Vortex Methods – Theory and Practice*. New York: Cambridge University Press; 2000.
- [19] Peskin C. Flow patterns around heart valves: a numerical study. *J Comput Phys* 1972;10:252–71.
- [20] Arthurs KM, Moore LC, Peskin CS, Pitmann EB, Layton HE. Modeling arteriolar flow and mass transport using the immersed boundary method. *J Comput Phys* 1998;147:402–40.
- [21] Lai M-C, Peskin CS. An immersed boundary method for formal second-order accuracy and reduced numerical viscosity. *J Comput Phys* 2000;160:705–19.
- [22] Leveque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficient and singular sources. *SIAM J Numer Anal* 1994;31(4):1019–44.
- [23] Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J Comput Phys* 2001;171:132–50.
- [24] Goldstein D, Handler R, Sirovich L. Direct numerical simulation of turbulent flow over a modelled riblet covered surface. *J Fluid Mech* 1995;302:333–76.
- [25] Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *J Comput Phys* 1993;105:354–66.
- [26] Cottet G-H, Maitre E. A level-set formulation of immersed boundary methods for fluid–structure interaction problems. *Math Model Meth Appl Sci* 2006;16(3):415–38.
- [27] Cottet G-H. Artificial viscosity models for vortex and particle methods. *J Comput Phys* 1996;127:299–308.
- [28] Koumoutsakos P. Inviscid axisymmetrization of an elliptical vortex ring. *J Comput Phys* 1997;138:821–57.
- [29] Degond P, Mas-Gallic S. The weighted particle method for convection–diffusion equations. Part 1: The case of an isotropic viscosity. *Math Comput* 1989;53(188):485–507.
- [30] Chorin AJ. Numerical study of slightly viscous flow. *J Fluid Mech* 1973;57(4):785–96.

- [31] Larsen A, Walther JH. Discrete vortex simulation of flow around five generic bridge deck sections. *J Wind Eng Ind Aerodyn* 1998;77–78:591–602.
- [32] Wu JC. Theory for aerodynamics force and moments in viscous flows. *AIAA J* 1981;19(4):432–41.
- [33] Koumoutsakos P, Leonard A. Improved boundary integral method for inviscid boundary condition applications. *AIAA J* 1993;31(2):401–4.
- [34] Wu JC. Numerical boundary conditions for viscous flow problems. *AIAA J* 1976;14(8):1042–9.
- [35] Kinney RB, Cielak ZM. Analysis of unsteady viscous flow past an airfoil: Part I – theoretical development. *AIAA J* 1977;15(12):1712–7.
- [36] Wu JC, Sampath S, Sankar NL. A numerical study of unsteady viscous flows around airfoils. In: AGARD fluid dynamics panel symposium on unsteady aerodynamics, Ottawa, Canada, vol. 227, AGARD, 1977. p. 24–1–18.
- [37] Taslim ME, Kinney RB. Analysis of two-dimensional viscous flow over cylinders in unsteady motion. *AIAA J* 1984;22(5):586–94.
- [38] Tuncer IH, Wu JC, Wang CM. Theoretical and numerical studies of oscillating airfoils. *AIAA J* 1990;28:1615–24.
- [39] Beale JT, Majda A. High order accurate vortex methods with explicit velocity kernels. *J Comput Phys* 1985;58:188–208.
- [40] Knio OM, Ghoniem AF. Three-dimensional vortex simulation of rollup and entrainment in a shear layer. *J Comput Phys* 1991;97:172–223.
- [41] Abdolhosseini R, Milane RE. Vortex-in-cell simulation of a spatially growing mixing layer. *Mech Res Comm* 1998;25(3):237–44.
- [42] Walther JH, Koumoutsakos P. Three-dimensional particle methods for particle laden flows with two-way coupling. *J Comput Phys* 2001;167:39–71.
- [43] Kuethe AM, Chow C. Foundation of aerodynamics. 4th ed. John Wiley and Sons; 1986.
- [44] Katz J, Plotkin A. Low-speed aerodynamics, from wing theory to panel methods. McGraw Hill, Inc.; 1991.
- [45] Hald OH. Convergence of a random method with creation of vorticity. *SIAM J Sci Stat Comput* 1986;7:1373–86.
- [46] Roberts S. Accuracy of the random vortex method for a problem with non-smooth initial conditions. *J Comput Phys* 1985;58:29–43.
- [47] Roberts S. Convergence of a random walk method for the Burgers equation. *Math Comput* 1989;52(186):647–73.
- [48] Roberts S. A particle method for a scalar advection diffusion equation. *Math Comp Sim* 1990;32:155–60.
- [49] Ghoniem AF, Sherman FS. Grid-free simulation of diffusion using random walk methods. *J Comput Phys* 1985;61:1–37.
- [50] Smith PA, Stansby PK. An efficient surface algorithm for random-particle simulation of vorticity and heat transport. *J Comput Phys* 1989;81:349–71.
- [51] Milinazzo F, Saffman PG. The calculation of large Reynolds number two-dimensional flow using discrete vortices with Random Walk. *J Comput Phys* 1977;23:380–92.
- [52] Shankar S, van Drommelen L. A new diffusion procedure for vortex methods. *J Comput Phys* 1996;127:88–109.
- [53] Carrier J, Greengard L, Rokhlin V. A fast adaptive multipole algorithm for particle simulations. *SIAM J Sci Stat Comput* 1988;9(4):669–86.
- [54] Baker GR. The “Cloud-in-Cell” technique applied to the roll up of vortex sheets. *J Comput Phys* 1979;31:76–95.
- [55] Brecht SG, Ferrante JR. Vortex-in-cell calculations in three dimensions. *Comp Phys Commun* 1990;58:25–54.
- [56] Cottet G-H, Michaux B, Ossia S, VanderLinden G. A comparison of spectral and vortex methods in three-dimensional incompressible flows. *J Comput Phys* 2002;175:702–12.
- [57] Hockney RW, Eastwood JW. Computer simulation using particles. 2nd ed. Bristol, PA, USA: Institute of Physics Publishing; 1988.
- [58] Walther JH. An influence matrix particle–particle–mesh algorithm with exact particle–particle correction. *J Comput Phys* 2003;184:670–8.
- [59] Monaghan JJ. Extrapolating B splines for interpolation. *J Comput Phys* 1985;60(2):253–62.
- [60] Hockney RW. The potential calculation and some applications. *Methods Comput Phys* 1970;9:136–210.
- [61] Walther JH. An influence matrix particle–particle particle–mesh algorithm with exact particle–particle correction. *J Comput Phys* 2003;184:670–8.
- [62] Anderson CR. A method of local corrections for computing the velocity field due to a distribution of vortex blobs. *J Comput Phys* 1986;62(1):111–23.
- [63] Theuns T. Parallel P3M with exact calculation of short range forces. *Comp Phys Commun* 1994;78:238–46.
- [64] Phillips JR, White JK. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Trans Comput-Aided Des Integrated Circuits Syst* 1997;16(10):1059–72.
- [65] Sbalzarini IF, Walther JH, Bergdorf M, Hieber SE, Kotsalis EM, Koumoutsakos P. PPM – a highly efficient parallel particle–mesh library for the simulation of continuum systems. *J Comput Phys* 2006;215:566–88.
- [66] Bathe KJ, Zhang H. Finite element developments for general fluid flows with structural interactions. *Int J Numer Methods Eng* 2004;60:213–32.
- [67] Morgenthal G. Advances in numerical bridge aerodynamics and recent applications. *Struct Eng Int* 2005;15:95–100.
- [68] Green D, Unruh WG. The failure of the Tacoma Bridge: a physical model. *Am J Phys* 2006;74:706–16.
- [69] Cottet G-H, Koumoutsakos P, Salihi MLO. Vortex methods with spatially varying cores. *J Comput Phys* 2000;162(1):164–85.
- [70] Perlman M. On the accuracy of vortex methods. *J Comput Phys* 1985;59:200–23.
- [71] Talley S, Iaccarino G, Mungal G, Mansour N. An experimental and computational investigation of flow past cacti. Annual research briefs. Center for Turbulence Research, Stanford 2001. p. 51–63.