JoT JOURNAL OF TURBULENCE

http://jot.iop.org/

An immersed interface method for the vortex-in-cell algorithm†

J H Walther[1]‡ and G Morgenthal[2]

[1] Institute of Computational Science, ETH Zürich, CH-8092 Zürich, Switzerland
[2] Department of Engineering, University of Cambridge,
Cambridge CB2 1PZ, UK
E-mail: walther@inf.ethz.ch and gm249@cam.ac.uk

Received 25 September 2002
Published 18 October 2002

Abstract. The paper presents a two-dimensional immersed interface technique for the vortex-in-cell (VIC) method for the simulation of flows past complex geometries. The particle–mesh VIC algorithm is augmented by a local particle–particle (PP) correction term in a particle–particle particle–mesh (PPPM) context to resolve sub-grid scales incurred by the presence of the immersed interface. The PP correction furthermore allows mesh and particle resolution to be disjoined by explicitly resolving sub-grid scales on the particles. This PPPM algorithm uses an influence matrix technique to annihilate the anisotropic sub-grid scales and an exact PP 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 stream function. 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. 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 and temporal parameters. Finally, the flow past a cactus-like geometry is considered, demonstrating the efficient handling of complex bluff body flows.

PACS numbers: 02.70.Ns, 47.32.Cc

† This article was chosen from Selected Proceedings of the 4th International Workshop on Vortex Flows and Related Numerical Methods (UC Santa-Barbara, 17–20 March 2002) ed E Meiburg, G H Cottet, A Ghoniem and P Koumoutsakos.

‡ Author to whom any correspondence should be addressed.

 PII: S1468-5248(02)54105-9 1468-5248/02/000001+9$30.00

JoT 3 (2002) 039

Contents

1. Introduction

Particle vortex methods are based on a Lagrangian formulation of the Navier–Stokes equation and a discretization of the vorticity field onto vortex elements. The key computational element in vortex methods is the classical N-body problem governing the mutual interaction of the N vortex elements. Fast multipole methods [1] and fast hybrid particle–mesh (PM) algorithms such as the vortex-in-cell (VIC) algorithm [2] have been devised, reducing the operational cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ or $\mathcal{O}(N)$ depending on the complexity of the algorithm. The VIC method is typically employed for problems in periodic systems and for simple geometries which allow the use of fast Fourier transforms (FFTs) for the solution of the Poisson equation relating the fluid vorticity and velocity fields.

The present study extends the classical VIC algorithm by including the immersed interface method for the simulation of flows past complex bodies. The technique involves:

(i) an efficient solution of the Poisson equation using a FFT solver,

(ii) an exact prescription of free-space boundary conditions using a minimal number of grid points [3], and

(iii) an automatic resolution of sub-grid scales through the application of a direct particle–particle (PP) correction term.

Results are presented for the impulsively started flow past a circular cylinder at Reynolds number 3000. The simulation of the flow past a cactus-like geometry is performed to demonstrate the flexibility of the proposed algorithm.

2. Particle vortex methods

The classical two-dimensional vortex method is based on the discretization of the vorticity field by a finite sum of N_p Lagrangian particles

$$\omega(\boldsymbol{x}, t) = \sum_{p=1}^{N_p} \eta_\sigma(|\boldsymbol{x} - \boldsymbol{x}_p|)\Gamma_p, \tag{1}$$

where $(\boldsymbol{x}_p, \Gamma_p)$ denotes the particle position and strength, respectively. The vorticity of each particle is mollified over a support of size σ by a smooth function $\eta_\sigma(\boldsymbol{x})$

$$\eta_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^2}\eta\left(\frac{|\boldsymbol{x}|}{\sigma}\right) \tag{2}$$

and $\eta(r)$ is the cut-off function. A second-order Gaussian [4] is used in the present study.

The velocity field in free space is determined from the Biot–Savart relation

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{U}_\infty - \frac{1}{2\pi}\sum_{p=1}^{N_p} \boldsymbol{K}_\sigma(\boldsymbol{x}-\boldsymbol{x}_p)\times\Gamma_p\boldsymbol{e}_z, \tag{3}$$

where the velocity kernel $\boldsymbol{K}(\boldsymbol{x}) = \boldsymbol{x}/|\boldsymbol{x}|^2$ has been convolved with the mollifier to obtain $\boldsymbol{K}_\sigma(\boldsymbol{x}) = \boldsymbol{K} * \eta_\sigma(\boldsymbol{x})$. The vortex particles evolve according to the vorticity transport equation and are solved using a fractional step algorithm [5]. The inviscid step of the algorithm is approximated by the convection of the vortex elements. The diffusion problem is solved using the random walk technique [6]. A boundary element method (BEM) is employed for the description of the immersed interface and enforcement of the kinematic boundary condition $\boldsymbol{u}\cdot\boldsymbol{n}|_{\mathcal{B}} = 0$, where $\mathcal{B}$, denotes the solid boundary. A second-order method is obtained by assuming a linear variation of the surface vortex sheets (γ_i). During the diffusion step, each of the vortex sheets are converted into a number of vortex elements, which are subsequently diffused into the flow using one-sided random walks. Particles diffusing back into the solid are deleted [7], and their strength and position are recorded for the calculation of the total flux of vorticity.

Thus the present random vortex algorithm involves the following steps:

(i) solve for the vortex sheet strength by enforcing the kinematic boundary condition using the BEM;

(ii) compute the particle velocity ($\boldsymbol{u}_p$) using equation (3) or the PPPM algorithm described in the following section;

(iii) convect the particles to solve the inviscid problem;

(iv) convert the vortex sheets into vortex elements;

(v) solve the diffusion problem using the random walk technique.

3. Hybrid particle–mesh algorithms

Fast algorithms for the solution of the N-body problem include the VIC technique [2] typically employed to problems in simple geometries and for periodic systems. Recent extensions to the algorithm include domain decomposition techniques and local body fitted meshes to allow treatment of more complex geometries such as the flow past multiple circular cylinders, compare with [8, 9].

An alternative approach that avoids the generation of curvilinear meshes is the immersed interface method by Peskin [10]. Here the boundary conditions are enforced on the immersed interface through source terms acting on a regular mesh. The present work combines the immersed interface method of Peskin with the VIC algorithm using a boundary element description of the interface. However, while the VIC method generally provides an accurate solution for smooth vorticity fields, the discontinuity introduced at an immersed interface results in sub-grid scales not resolved by the VIC method. Thus, an influence matrix particle–particle particle–mesh algorithm (PPPMi) developed for $1/r$ potentials [11] has been adapted for the two-dimensional particle vortex method as described in the following sections.

3.1. The vortex-in-cell algorithm

The VIC algorithm attains its efficiency by employing fast Fourier or multigrid methods for the solution of the Poisson equation for the stream function ($\nabla^2\Psi = -\omega$) on a mesh. The vorticity

field is constructed on the mesh from the strength of the particles using a smooth projection

$$\omega_m(\boldsymbol{x}_m) = \frac{1}{h^2}\sum_{p=1}^{N_p} W(\boldsymbol{x}_p - \boldsymbol{x}_m)\Gamma_p, \tag{4}$$

where $W(\boldsymbol{x})$ denotes the interpolation kernel and h is the mesh spacing. The third-order M_4' kernel proposed by Monaghan [12] is used in the present study

$$M_4'(x) = \begin{cases} 1 - \frac{5}{2}|x|^2 + \frac{3}{2}|x|^3, & |x| \le 1, \\ \frac{1}{2}(2-|x|)^2(1-|x|), & 1 < |x| < 2, \\ 0, & |x| \ge 2. \end{cases} \tag{5}$$

After the projection step, the Poisson equation is solved using FFTs with the method proposed by Hockney [3] to recover the exact free-space boundary condition. The procedure involves an extension of the computational domain in the free-space direction(s) and a modification of the Green function (G) on the mesh according to

$$\left.\begin{array}{l} G_{i,j} = (i^2+j^2)^{-1/2}, \\ G_{2N_x-i,j} = G_{i,2N_y-j} = G_{2N_x-i,2N_y-j} = G_{i,j}, \end{array}\right\} \begin{array}{l} 0 \le i \le N_x, \\ 0 \le j \le N_y, \\ i+j \ne 0, \end{array}$$
$$G_{0,0} = 1, \tag{6}$$

where (i,j) denotes mesh point and $(2N_x, 2N_y)$ the number of mesh points in the extended system. The velocity field is computed from the stream function on the mesh

$$\boldsymbol{u} = \boldsymbol{\nabla} \times \Psi \boldsymbol{e}_z, \tag{7}$$

using fourth-order finite differences. The resolved velocity field is finally interpolated back onto the particles

$$\boldsymbol{u}_p = \sum_{m=1}^{N_x N_y} W(\boldsymbol{x}_p - \boldsymbol{x}_m)\boldsymbol{u}_m. \tag{8}$$

3.2. The particle–particle particle–mesh algorithm

The VIC algorithm described in section 3.1 requires that equation (4) is a close approximation to the exact vorticity field. This is generally true for smooth vorticity fields, whereas the discontinuity introduced by the immersed interface results in a particle velocity field $\boldsymbol{u}_p$ that contains sub-grid scales $\boldsymbol{u}_{pp}$,

$$\boldsymbol{u}_p = \tilde{\boldsymbol{u}}_p + \boldsymbol{u}_{pp}, \tag{9}$$

where $\tilde{\boldsymbol{u}}_p$ is the grid-resolved velocity. The magnitude and distribution of these sub-grid scales will generally depend on the specific particle field and the interpolation kernel used for the projection steps (equations (4) and (8)). The limited number of statistical moments preserved by the kernel and the directional sensitivity of the differential operators applied to equation (7) lead to an anisotropic field of sub-grid scales not resolved in the PM technique as shown in figure 1. The figure compares the induced velocity of a vortex element computed using the VIC algorithm for different relative positions on the mesh. The VIC results are in good agreement with the exact solution for $x/h > 4$, but display a strong anisotropy for $x/h < 4$.

The accuracy of the direct PP interaction and the efficiency of the PM algorithm have been successfully combined in the hybrid PPPM algorithm. While the original method devised by Hockney and Eastwood [13] relies on an optimization of the Green function in Fourier space to enforce a prescribed radially symmetric sub-grid scale $\boldsymbol{u}_{pp}$. the present algorithm allows iterative

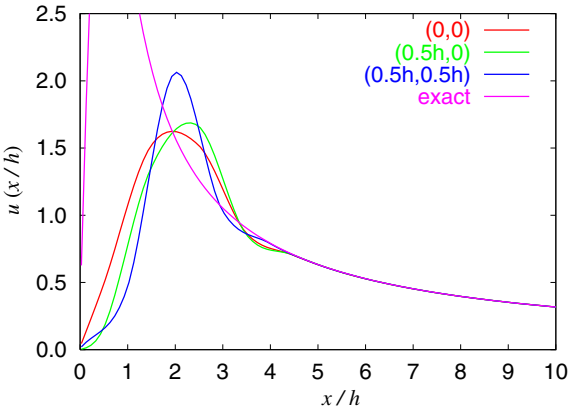

Figure 1. The induced velocity (u) computed using the VIC algorithm of a vortex element located at different positions relative to the VIC mesh. The exact smooth velocity field is given by equation (3).

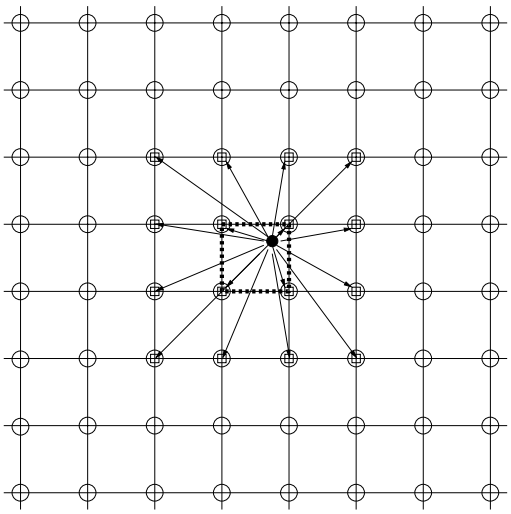

Figure 2. Schematic diagram of the PPPM algorithm for the $M'4$ projection kernel and two neighbouring cells ($K = 2$) for the PP correction. The particles (●) contained within the dashed box contribute to the vorticity field of the (4×4) mesh points marked with a box (□). The mesh points included in the influence matrix are marked with a circle (○).

solvers using finite differences through an influence matrix PPPM technique (PPPMi) [11]. The algorithm is similar to the method of local corrections by Anderson [14] and Theuns [15] but uses an influence matrix technique to provide an exact estimate for the resolved field on the mesh. 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 grid box is computed as

$$\boldsymbol{\delta u}_m = \mathsf{C}\boldsymbol{\delta\omega}_m, \tag{10}$$

wherein $\boldsymbol{\delta u}_i$ is the velocity of the ith grid point induced by a nodal vorticity $\delta\omega_j$ at the jth grid point. This influence is localized over L grid points, the near-range region of each grid box (see figure 2).

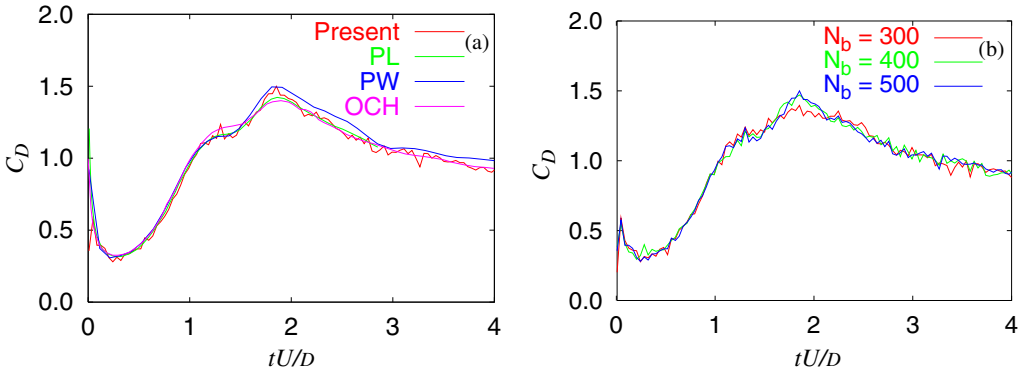

Figure 3. Time history of the drag for the impulsively started flow past a circular cylinder at Reynolds number 3000. The present results are compared in (a) with previous vortex method simulations. The dependence of the number of boundary elements is shown in (b).

The influence matrix then has the dimensions $L \times M$, where M is the kernel's number of supporting nodes and the dimension of vector $\boldsymbol{\delta\omega}_m$. For the M_4' kernel and K neighbouring grid boxes included in the PP correction $M = 4 \times 4$ and $L = (2K + 4)^2$, compare with figure 2. The velocity field computed by equation (10) is subsequently subtracted on the respective grid points, relying on the linearity of the Poisson equation to cancel exactly the PM contribution of the particles contained in the originating grid box. In the following PP step an exact (equation (3)) velocity correction between the particles within the neighbourhood provides the final velocity prediction of the PPPMi algorithm.

The influence matrix is constructed by placing M individual test particles of unit strength at arbitrary (but different) positions in a central grid box of the domain and evaluating the corresponding PM-predicted velocity field. The projected vorticity of the kth test particle on the mth grid point $\delta\Omega_{m,k}$ is determined using (4). Next, the velocity field is computed using the Poisson solver of the PM part and sampled at the L grid points neighbouring the sample grid box as $\boldsymbol{\delta U}_{i,k}$. Assembling this for the M test particles forms the following linear system of equations ($\boldsymbol{\delta U}_{i,k} = \mathsf{C}_{i,m}\delta\Omega_{m,k}$) which is solved for the elements of C during the initialization step of the algorithm. Summarizing, the influence matrix PPPM algorithm proceeds as follows:

(i) project the particle strengths Γ_p onto the mesh using equation (4) and store the contributions separately;

(ii) solve the Poisson equation $\nabla^2\Psi = -\omega$ on the mesh;

(iii) compute the velocity field $\boldsymbol{u} = \boldsymbol{\nabla} \times \Psi \boldsymbol{e}_z$ on the mesh;

(iv) interpolate the resolved velocity field back onto the particles to obtain $\tilde{\boldsymbol{u}}_p$;

(v) for each non-empty grid box perform the following steps to solve for its PM-resolved velocities at the neighbouring grid boxes induced by its projected vorticity $\boldsymbol{\delta\omega}_m$:

  (a) compute the velocity at the neighbouring grid nodes by applying the influence matrix (equation (10)): $\boldsymbol{\delta u}_m = \mathsf{C}\boldsymbol{\delta\omega}_m$;

  (b) for each of the neighbouring grid boxes project the resulting velocity onto the particles to obtain $\boldsymbol{u}_{ps}$;

(vi) compute the sub-grid velocities as a local PP interaction correction $\boldsymbol{u}_{pc}$;

(vii) contributions from any external velocity potentials are included as $\boldsymbol{u}_{pe}$;

(viii) the total particle velocity is $\boldsymbol{u}_p = \tilde{\boldsymbol{u}}_p - \boldsymbol{u}_{ps} + \boldsymbol{u}_{pc} + \boldsymbol{u}_{pe}$.

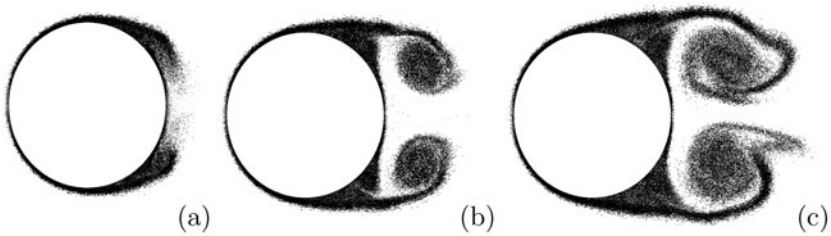

Figure 4. Snapshots of the particle map for the impulsively started flow past a circular cylinder at Reynolds number 3000 and $tU/D = 1.0$ (a), 2.0 (b) and 3.0 (c).

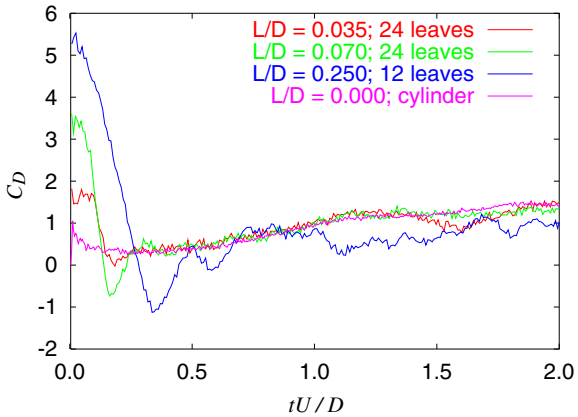

Figure 5. Time history of the drag coefficient for the impulsively started flow past cacti at different corrugations.

4. Results

The first test case involves the impulsively started flow past a circular cylinder at Reynolds number 3000. The present results are compared with recent simulations of Koumoutsakos and Leonard (KL) [16], Ould-Salihi, Cottet and El Hamraoui (OCH) [9] and Ploumhans and Winckelmans (PW) [17]. The spatial resolution involves 400 boundary elements and a maximum of six particles created at each of the elements. The simulations extend to a non-dimensional time of $tU/D = 4$, with a time step of $\delta tU/D = 0.004$. The drag coefficient is presented without any filtering and is found to be in good agreement with previous results (see figure 3(a)). The spatial convergence of the proposed algorithm is considered by varying the number of boundary elements as shown in figure 3(b). The simulation using 300 elements clearly underestimates the maximum drag value, where the 400 and 500 cases appear to have converged. Snapshots of the particle map at $tU/D = 1, 2$, and 3 are shown for the reference case in figure 4.

The second study involves the impulsively started flow past a cactus-like geometry [18]. Three different geometries are considered: two cacti with 24 leaves and cavity depths of $L/D = 0.035$ and 0.07, respectively, and one cactus with 12 leaves and a cavity depth of 0.25. The simulations are performed at a Reynolds number of 3000, and conducted until a non-dimensional time of 2 with a time step of 0.0015. The number of boundary elements is chosen such as to obtain panel lengths of approximately $\delta s/D = 0.003$, corresponding to 1080,

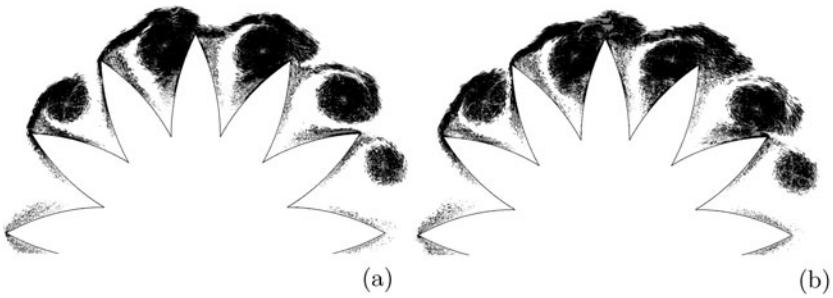

Figure 6. Snapshots of the particle map for the impulsively started flow past a 12 leaf cactus with $L/D = 0.250$ at a Reynolds number of 3000. The snapshots correspond to the time of the first (a) and second (b) occurrence in the drag force at $tU/D = 0.034$ and 0.570 respectively.

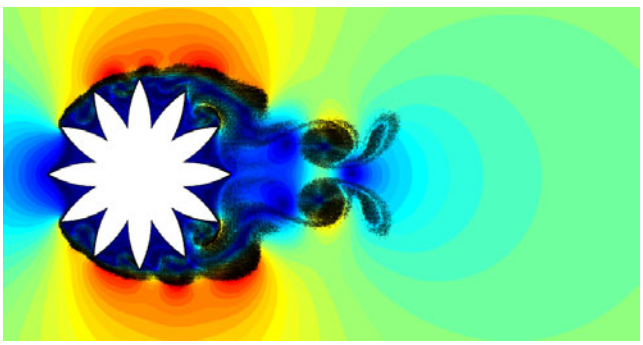

Figure 7. Snapshots of the particle map superimposed on contours of the fluid speed at $tU/D = 2.00$ for the 12-leaf case shown in figure 6. Animation of particle evolution and fluid speed (3 MB AVI).

1440 and 1920 elements for the three cases. Partial remeshing [19] is used at every time step, with a particle mesh resolution of $h_p/\delta s = 0.5$. The VIC mesh spacing is $h/h_p = 10.6$, resulting in a mesh size of $(256, 128)$ at the end of the simulation. The range of local PP correction is $K = 4$.

The time histories of the drag force obtained for the different geometries are compared in figure 5 with the drag force for the circular cylinder. At early times ($tU/D < 0.1$) the corrugated cylinders exhibit a marked increase in the drag of 1.6, 3.5 and 5.3 compared to a value of approximately 1.0 for the circular cylinder. This increased initial drag is a result of the formation of the vortices at the tip of the spines. As these vortices detach, the drag decreases and reaches a minimum as the vortices touch the downstream spine. The low surface pressure in the vicinity of the vortices is responsible for the observed negative drag. As the vortices diffuse and convect out of the cavity the drag recovers to a value in close agreement with that of the circular cylinder. However, the 12 leaf case experiences a second drag minimum as the vortices in the upstream pointing cavities touch their downstream spine wall at approximately $tU/D = 0.57$. Snapshots of the particle map at these minima are shown in figure 6. Only particles with a strength (Γ_p) exceeding 2×10^{-5} are shown, thus only displaying 25% of the particles present. The final flow pattern at $tU/D = 2.00$ is shown in figure 7. The wake is narrow, consisting of two colliding dipoles. The colour coding indicates the local fluid speed.

5. Summary and conclusions

An immersed interface PPPM algorithm has been presented for the simulation of two-dimensional bluff body flows. The PM step of the algorithm extends the VIC 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 PP step involves an exact PP correction term. The PPPM algorithm furthermore allows us to disjoin the mesh and particle resolution by explicitly resolving sub-grid scales on the particles. The simulations of the impulsively started flow past a circular cylinder at Reynolds number 3000 demonstrated the convergence of the proposed methodology. Finally, the flow past a cactus-like geometry was conducted to demonstrate the ability of the method to handle complex geometries.

References

[1] Greengard L and Rokhlin V 1987 A fast algorithm for particle simulations *J. Comput. Phys.* **73** 325–48
[2] Christiansen J P 1973 Numerical simulation of hydrodynamics by the method of point vortices *J. Comput. Phys.* **13** 363–79
[3] Hockney R W 1970 The potential calculation and some applications *Methods Comput. Phys.* **9** 136–210
[4] Beale J T and Majda A 1985 High order accurate vortex methods with explicit velocity kernels *J. Comput. Phys.* **58** 188–208
[5] Chorin A J 1978 On the convergence of discrete approximations to the Navier–Stokes equations *SIAM J. Sci. Stat. Comput.* **27** 341–53
[6] Chorin A J 1973 Numerical study of slightly viscous flow *J. Fluid Mech.* **57** 785–96
[7] Smith P A and Stansby P K 1989 An efficient surface algorithm for random-particle simulation of vorticity and heat transport *J. Comput. Phys.* **81** 349–71
[8] Slaouti A and Stansby P K 1992 Flow around two circular cylinders by the random-vortex method *J. Fluids Struct.* **6** 614–70
[9] Ould-Salihi M L, Cottet G-H and El Hamraoui M 2000 Blending finite-difference and vortex methods for incompressible flow computations *SIAM J. Sci. Comput.* **22** 1655–74
[10] Peskin C 1972 Flow patterns around heart valves: a numerical study *J. Comput. Phys.* **10** 252–71
[11] Walther J H 2002 An influence matrix particle–particle particle–mesh algorithm with exact particle–particle correction *J. Comput. Phys.* at press
[12] Monaghan J J 1985 Extrapolating B splines for interpolation *J. Comput. Phys.* **60** 253–62
[13] Hockney R W and Eastwood J W 1988 *Computer Simulation Using Particles* 2nd edn (Bristol: Institute of Physics Publishing)
[14] Anderson C R 1986 A method of local corrections for computing the velocity field due to a distribution of vortex blobs *J. Comput. Phys.* **62** 111–23
[15] Theuns T 1994 Parallel P3M with exact calculation of short range forces *Comput. Phys. Commun.* **78** 238–46
[16] Koumoutsakos P and Leonard A 1995 High-resolution simulation of the flow around an impulsively started cylinder using vortex methods *J. Fluid Mech.* **296** 1–38
[17] Ploumhans P and Winckelmans G S 2000 Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry *J. Comput. Phys.* **165** 354–406
[18] Talley S, Iaccarino G, Mungal G and Mansour N 2001 An experimental and computational investigation of flow past cacti *Annual Research Briefs (Center for Turbulence Research, Stanford 2001)* pp 51–63
[19] Morgenthal G and Walther J H 2002 An immersed interface method for the vortex-in-cell algorithm, in preparation