Available online at www.sciencedirect.com
ScienceDirect
Computers and Structures 85 (2007) 712–726
Computers & Structures
www.elsevier.com/locate/compstruc

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 Particle–Mesh (P^3M) 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 P^3M 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 P^3M 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.

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.

0045-7949/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2007.01.020

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 particle–mesh (P^3M) 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 P^3M 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 P^3M 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 $\mathscr{D}$ bounded by $\partial\mathscr{D} \equiv \mathscr{B}$ is governed by the vorticity transport equation

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

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

$$\boldsymbol{u}(\boldsymbol{x}) \to \boldsymbol{U}_\infty \quad \text{as } |\boldsymbol{x}| \to \infty, \tag{2}$$

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

$$\nabla^2 \Psi = -\omega. \tag{3}$$

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