Comparison between Vortex-In-Cell and Vortex Particle Methods on Two-Dimensional Problem

C X Canh¹, L R Zuhal²*, and H Muhammad³

¹ Modeling & Simulation Center, Viettel High Technology Industries Corporation, 380 Lac Long Quan, Nhat Tan, Tay Ho, Hanoi, Vietnam
² Faculty of Mechanical and Aerospace Engineering, Bandung Institute of Technology, Ganesha No.10, 40132, Bandung, Indonesia

* Correspondence: lavirz@ae.itb.ac.id
Received: 14 April 2021; Accepted: 26 May 2021; Published: 30 June 2021

Abstract: This research is concerned with the two-dimensional vortex method (VM) solvers. We develop and investigate the performance of the Vortex-In-Cell (VIC) and Vortex Particle Method (VPM) which are well known as the VM’s family members. The advantage of these both methods are that we can accelerate velocity computation procedure, an N-body problem in numerical methods, by using Fast Fourier Transform (FFT) and Fast Multipole Method (FMM), respectively. In addition, the viscous calculation process in VPM can be accelerated by using a scheme of Nearest Neighbor Particle Searching (NNPS) algorithms. Moreover, the no-through boundary condition treatment issue can be easily handled by using an immersed boundary condition for both methods. The accuracy and numerical cost of both numerical methods will be examined by simulating flow over an Impulsively Started Circular Cylinder and comparisons.

Keywords: vortex methods, fast Fourier transform, fast multipole method, boundary condition

1. Introduction

Today, computational fluid dynamics (CFD) methods have been used extensively for solving fluid mechanics, besides the more expensive experiments. In CFD, the physical domain is discretized by using mesh/grid, cells, nodes or particles generation. However, the methods based on a grid are required to have a high grid generation cost for complex, deforming, and moving bodies. The meshfree methods which established on particles are also meet defective aspects, especially, solving N-body problems frequently requires (N²) order of time computation. These may become computationally expensive.

Hence, the improvement of acceleration method is one of most important preconditions to accelerate the computational time and provide accurate results. In this research, we present the Vortex-In-Cell (VIC) method [1,3] and Vortex Particle Method (VPM) [4,5] as the members of the Vortex Methods (VMs) family. The hybrid VIC algorithm is employed in this study which is presented and concluded that the grid-based and meshfree method is blended as the blending algorithms particle-mesh/grid methods. The VIC interpolates the particle strength to an underlying mesh. This method has the advantage that the Poisson inversion can be accomplished and accelerated by Fast Fourier transform techniques [3].

For the comparison, we also use a meshless fluid computational solver Fast Lagrangian Vortex Particle Method two-dimensional code which developed by others [4-5]. The method employs regularized particles that have been introduced by Winckelmans, 1993 [6], and Ploumhans, 2000 [7]. The classical Vortex Particle Method which employs Greens function to solve the Poisson equation is
accelerated as well, by using Fast Multipole Method (FMM). Further, the vorticity strength of particles will be diffused by diffusion modeling Particle Strength Exchange (PSE) which can be speeded up by utilization of NNPS scheme.

Additionally, boundary condition treatment is big trouble for all researchers who are in charge with fluid flow past the immersed obstacle even in both grid-based or meshfree methods. In the long journey of seeking the boundary condition model to simulate the bounded flow, we applied the existing higher order Brinkman penalization into grid field of VIC method to overcome this difficulty. For particles of VPM, we also develop an immersed boundary condition model based on Brinkman penalization for the enforcement of the no-slip boundary condition.

The objective of this work is to numerically investigate the performance of VIC and VPM in VMs family. In particular, our study focuses on the speed of FFT, FMM, NNPS schemes and their own accuracy.

2. Materials and Methods

2.1. Vortex Methods—Governing equation

The main goal of almost CFD is the numeric approximation by the Navier-Stokes equations that govern for fluid flow. In the two-dimensional VMs, such as 2D-VIC and 2D-VPM, they all applied the Navier-Stokes equations in form of velocity-vorticity formulation as follows:

\[
\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega + \nu \nabla^2 \omega = \text{B.C}
\]

(1)

where an immersed boundary condition (B.C) that based on Brinkman penalization can be enforced alternatively solid surfaces as the Dirichlet boundary condition, a detailed description can be found somewhere else [3]. The vorticity \( \omega = u \) is a tendency of a fluid element to spin. Moreover, the Helmholtz decomposition theorem shows that vector field \( u \) can be decomposed into rotational and irrotational part, \( u = \psi + \varphi = \psi + U_\infty \). Combining these equations we get the Poisson equation

\[
\nabla^2 \psi = -\omega
\]

(2)

where \( \psi \) is the associated stream function and \( U_\infty \) is irrotational velocity potential.

The algorithm for solving was first formally imposed into vortex method by Chorin, 1973 [2] as a necessary tool to resolve the vorticity-velocity equation 1 algorithmically and numerically. In the algorithm convection and diffusion terms will be separated fractionally into two sub-steps. The 1st sub-step is supposed that particle is moving on the domain using Lagrangian convection.

\[
\frac{d\omega}{dt} = \frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = 0; \quad \frac{d\mathbf{x}}{dt} = \mathbf{u}
\]

(3)

Then, in the 2nd sub-step the evaluating vorticity is implemented on the grid by calculating diffusion without particles movement

\[
\frac{d\omega}{dt} = \frac{\partial \omega}{\partial t} + \mathbf{0} \cdot \nabla \omega = \nu \nabla^2 \omega; \quad \frac{d\mathbf{x}}{dt} = \mathbf{u} = 0
\]

(4)
2.2. Vortex In Cell

The VIC scheme was introduced in 1973 by Christiansen [1]. This method utilizes grid (Eulerian) to solve the governing equation of the fluid and move them as a particle (Lagrangian). That means a Eulerian grid can be implemented in order to compute efficiently the velocity field on the Lagrangian particles. Moreover, this Eulerian grid can be used in order to compute diffusion and baroclinic terms in the governing equations, and particularly a grid-based Poisson solver for the calculation of particle velocities. The connection between two parts of that will be supported by the using values interpolation scheme from the grid to particle and vice versa.

Discretization Our system is discretized by a hybrid particle-grid method. In order to solve governing equations of the fluid, we used finite difference techniques on an underlying mesh (Eulerian), particularly for the calculation of particle velocities. Then, move them as a particle (Lagrangian). The relation between grid and particles is provided by interpolation scheme both from particle to mesh and mesh to the particle. The Neumann boundary condition is used for far-field boundary condition.

Velocity computation in order to obtain velocity field, solving Poissons equation (2) is probably the most important part of the whole VIC algorithm. The Poisson is an elliptic partial differential equation, based on the grid discretization, it can be solved by using iterative finite difference methods (FDM) instead of the high-cost Biot-Savart integration (O(N^2)). In our work, we used PSOR (point successive over-relaxation), it costs about O(N 3/2), and a cheaper method is known as FFT with an order of O(NlogN).

Diffusion On the discretization frame, the diffusion term (a parabolic partial differential equation) ∂ω/∂t = ν 2 ω is evaluated using finite difference schemes. The time discretization, which lies on the left-hand side (LHF), will only use the forward difference scheme. Other researcher explained two ways may be used, the first is the simple Euler time integration (first order), and the second is half-time integration (second order) [8].

Convection This is the Lagrangian aspect of this hybrid method. The particle will be moved to its new position. In equation 4, dx/dt = u shows that the particle is moved by integrating velocity over the time, and dω/dt = 0 shows that the vorticity contained by fluid particle remains constant during motion. In order to update the position of particles, we consider using Euler integration first order or second order scheme which Rosinelli 2011 [8] had given a clear description of how to evaluate the new particle position. After the particle moved out from the grid, we perform particle re-initialization every time step a by using a remeshing scheme.

2.3. Vortex Particle Method

In order to compare to VIC, in this research, we use a meshless fluid computational solver Fast Lagrangian Vortex Method three-dimensional code which developed by Zuhal 2014 [5] and Duong 2015 [4]. The method employs regularized particles that have been introduced by Winckelmans and Leonard 1993 [6], and Ploumhans and Winckelmans 2000 [7].

Discretization The essence of vortex particle method is an approximation of continuous vorticity field by discretization using the set of particles. Assume that a particle p is located at position xp, and it carries the information of fluid within a fluid volume Vp associated to particle p. In the case of the vortex method, the information is vorticity ω. Thus, by taking the integration of this quantity, we can easily estimate the strength of element over its volume (circulation times length), as follow:

\[ \Gamma = \int_{V_p} \omega dV = \omega \sigma^N, \quad 2D : \quad V_p \approx \sigma^2 \]  

where σ is core-size of the particle, which acts as a smoothing parameter, Γ is vorticity strength of each particle over its volume, with is the dimension of simulation.
Therefore, the two dimensional Navier-Stokes can be written under vortex strength-velocity form the equation (6) as follow:

$$\frac{d\Gamma}{dt} = \frac{\partial\Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma = \nu \int_{V_p} (\nabla^2 \omega) \, dV + \mathbf{B} \cdot \mathbf{C}$$

(6)

where the second term in LHS is convection, and the first term in RHS is the viscous term. Obviously, by using a splitting scheme that mentioned in subsection 2.1, the equation (6) can be solved. As discussed before in the previous subsection, for VIC method, must be modified to enforce far-field boundary conditions. However, VEM is using Biot-Savart relation which based on Greens green function approach inherently satisfies the far-field boundary condition.

Velocity computation in the convection sub-step 1, the position \( \mathbf{x} \) is tracked by their velocity which can be evaluated by Biot-Savart law as following equation:

$$u(x, t) = -\frac{1}{2\pi} \sum_{p} \frac{\rho_p \mathbf{n}_p}{|\mathbf{x} - \mathbf{x}_p(t)|^2} \times \mathbf{e}_z \Gamma_p(t)$$

(7)

where \( \sigma \) is the velocity smoothing kernel. We can directly calculate the velocity by using direct pairwise interactions which presented in equation (7). However, due to \( (N^2) \) order of time computation the computational cost is very expensive if a huge number of particles. In order to reduce computational cost, we following the FMM in other literatures [4,5].

Diffusion In the vortex element method, several schemes can be employed to resolve the vorticity evolution due to viscous diffusion, such as Core Expansion, Particle Strength Exchange, and Redistribution Method. Each scheme has own advantages and disadvantages. Following Zuhal 2014 [5], PSE scheme is selected due to the high accuracy and efficiency. However, PSE is known as a N-body problem. In this research, to shorten the computational time of estimating diffusion procedure we employed a scheme of Nearest neighbor particle searching (NNPS). Convection In this research, we used Euler integration first order \( x_{n+1} = x_n + \Delta t \mathbf{u} \).

3. Results and Discussion

In this section, for comparison between VIC and VPM, we consider the problem that presented in Cao 2016 [3], the impulsively started flow past a circular cylinder at \( \text{Re} = 550 \). Assume that the cylinder is initially at rest and surrounded by fluid, and then the fluid moves impulsively at a certain velocity. For VIC’s configuration Cao 2016 [3] used grid spacing equals to 0.025.

![Figure 1](image_url). History of \( \text{Cd} \) at \( \text{Re} = 550 \), \( \Delta t = 0.005 \), resolution 0.025, domain [16R 8R], compare with reference
Figure 2. History of Cd at Re = 550, dt = 0.001, resolution 0.01, domain [30R 20R], compare with reference

Table 1. Computation cost for T=5 simulation. On sever of 24 processors: Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz/Core i7

<table>
<thead>
<tr>
<th>Cases</th>
<th>Resolution</th>
<th>dt</th>
<th>Number of particles</th>
<th>Time cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPM-pure</td>
<td>σ = 0.025</td>
<td>0.005</td>
<td>1449 → 12590</td>
<td>1hour 28mins</td>
</tr>
<tr>
<td>VPM-FMM</td>
<td>σ = 0.025</td>
<td>0.005</td>
<td>1449 → 12590</td>
<td>18mins</td>
</tr>
<tr>
<td>VPM-FMM&amp;NNPS</td>
<td>σ = 0.025</td>
<td>0.005</td>
<td>1449 → 12590</td>
<td>12mins</td>
</tr>
<tr>
<td>VIC-PSOR</td>
<td>h = 0.025</td>
<td>0.005</td>
<td>321 × 161 = 51681</td>
<td>2hours 17mins</td>
</tr>
<tr>
<td>VIC-FFTW</td>
<td>h = 0.025</td>
<td>0.005</td>
<td>321 × 161 = 51681</td>
<td>2mins 36s</td>
</tr>
</tbody>
</table>

The interesting scope of domain is [16R 8R], time integration dt = 0.005. Thus, the number of particles to be 321 161 = 51681 particles. The stop condition for this iteration method is using iterative error s = 10–5 or maximum 1500 interaction for each time step. For VPM running, we also take dt = 0.005 and particle core-size σ = 0.025. The VPM’s results will be analyzed with VIC’s results in Cao 2016 [3] and reference’s results of Ploumhans 2000 [7] and Huang 2009 [9]. The table 1 shows the cases’ cost with the same running configuration: dt = 0.005, resolution = 0.025, using Euler integration first order convection.

The VPM-pure solves the Biot-Savart directly, then the time cost is quite expensive due to (N 2) order of Biot-Savart and viscous term. The VPM is accelerated by FMM (tree code (NlogN )) and NNPS ( (N )), but still keep the same result, see figure 1. Although the order of PSOR is (N 3/2) but the number of particles 51681 is bigger than 1449 12590 particles of VPM, hence, it is the costliest. The accurate of VIC-PSOR depends on the iterative error and order of FDM. Computation might be costly if the iteration cannot reach that criterion value due to too small iterative error or for some reasons, such as complicated vorticity fields. The VIC using FFT gives agreed result compare to using PSOR, see figure 1.

It can be seen that the fastest method is VIC using FFT. VIC method is also speeded up by a feature of VPM, the particles with ω > 0.001 ωmax will be re-meshed, refer to other results [7]. It
reduces the computational time for the case of VIC- FFTW from 8 minutes to 2mins36s. The figure 1 shown that, in the beginning, VIC quite closes to reference; however, after long and stable simulation, the VPM is the one that agrees with reference. The number of particles in VIC does not change during simulation.

Besides, VIC requires a big domain to obtain the acceptable result. Therefore, memory allocation for solving and saving data is the main problem if large simulation, especially for long-time simulation. And this leads to expensive postprocessing as well. In VPM, the number of particles is small at the beginning and increase after time, then this can save the computation for VPM.

Figure 2 shows agreement with reference when we increase the resolution and make dt finer. However, the VIC requires a bigger domain size to get better results. From figure 3, it can be seen that the simulation performed using the current work agrees with the reference Huang 2009 [9]. The shape of the blobs is matching at the three sample times of two methods with reference

4. Conclusions

In the present paper, the capability of the both VIC and VPM, are investigated by simulating two-dimensional problems of flow over an impulsively started circular cylinder. It has been shown that the present results are in good agreement with references. The VIC with FFT is the cheapest solution. However, the memory expense and accuracy of VIC are things need to be considered. Using VIC with finer resolution requires a larger domain size to get better results. This requirement may lead VIC to spend more expensive random-access memory, storage and computational cost than VPM. As discussed above, the Poisson and diffusion solvers in VIC need the enforcement of far-field boundary condition. If the incorrect far-field B.c. is applied, it can lead to the incorrect result, while VPM satisfies this requirement automatically via Biot-Savart. The VPM is even a higher cost solution but it saves the running memory and storage. Besides, the speed of meshless VPM could be increased by performing remeshing every several time-step instead of remeshing every time-step of the VIC

Acknowledgments
The 1st author would like to acknowledge the financial support from the Faculty of Mechanical and Aerospace Engineering Institut Teknologi Bandung.

References
8. Rossinelli D 2011 Eth Zurich, Diss. ETH No. 19575.