اونيۏرسيتي تيكنولوڬي مارا
UNIVERSITI TEKNOLOGI MARA

Journal of Mechanical Engineering

An International Journal

Volume 12 No. 1 | June 2015 | ISSN 1823-5514

Investigation on Heat Transfer Characteristics of Ceramic Coated Piston Crown for a CNGDI Engines — Helmisyah Ahmad Jalaludin

Accuracy Improvement for Linear Tetrahedral Finite Element by Means of Virtual Mesh Refinement — Sugeng Waluyo

Two-Dimensional Fast Lagrangian Vortex Method for Simulating Flows around a Moving Boundary — Duong Viet Dung, Lavi R. Zuhal, Hari Muhammad

Simulation Analysis of the Effect of Temperature on Overpotentials in PEM Electrolyzer System — A.H. Abdol Rahim, Alhassan Salami Tijani, Farah Hanun Shukri

The Effect of Skin Orientation on Biomechanical Properties — Nor Fazli Adull Manan, Jamaluddin Mahmud

A Study of Single and Two-Plane Balancing Using Influence Coefficient Method — Wan Sulaiman Wan Mohamad, A. A. Mat Isa, M.A Ismail

JOURNAL OF MECHANICAL ENGINEERING (JMechE)

Journal of Mechanical Engineering (ISSN 1823-5514) is published by the Faculty of Mechanical Engineering (FKM) and UiTM Press, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.

Journal of Mechanical Engineering

An International Journal

Volume 12 No. 1 **June 2015** **ISSN 1823-5514**

1. Investigation on Heat Transfer Characteristics of Ceramic Coated Piston Crown for a CNGDI Engines 1
   Helmisyah Ahmad Jalaludin

2. Accuracy Improvement for Linear Tetrahedral Finite Element by Means of Virtual Mesh Refinement 19
   Sugeng Waluyo

3. Two-Dimensional Fast Lagrangian Vortex Method for Simulating Flows around a Moving Boundary 31
   Duong Viet Dung
   Lavi R. Zuhal
   Hari Muhammad

4. Simulation Analysis of the Effect of Temperature on Overpotentials in PEM Electrolyzer System 47
   A.H. Abdol Rahim
   Alhassan Salami Tijani
   Farah Hanun Shukri

5. The Effect of Skin Orientation on Biomechanical Properties 67
   Nor Fazli Adull Manan
   Jamaluddin Mahmud

6. A Study of Single and Two-Plane Balancing Using Influence Coefficient Method 83
Wan Sulaiman Wan Mohamad
A. A. Mat Isa
M. A Ismail

Journal of Mechanical Engineering *Vol. 12, No. 1, 31-46, 2015*

Two-Dimensional Fast Lagrangian Vortex Method for Simulating Flows around a Moving Boundary

Duong Viet Dung
Lavi R. Zuhal
Hari Muhammad
Faculty of Mechanical and Aerospace Engineering
Bandung Institute of Technology, Jl. Ganesha, 10, Indonesia

ABSTRACT

This paper presents the development of an accelerated two-dimensional core spreading vortex method for simulating flows over a moving boundary. The complex geometry is treated as tracking particles, which are introduced within the extended fluid domain. The boundary conditions are enforced by generating wall vortex blobs at each time step based on representation of Nascent vortex elements. The viscous effect is modeled by core spreading method, with splitting and merging spatial adaptation scheme. The velocity field is calculated by using Biot-Savart formulation. In order to accelerate computation, the fast multipole method is also employed. The solver is validated by performing the simulations of flow around an impulsively moving cylinder at Reynolds number 550, and flow over a forced-oscillating flat plate at Reynolds number 10000. The results are found to be in good agreement with those reported in literatures.

Keywords: *Fluid structure interaction, vortex method, fast multipole method, splitting and merging, core spreading method*

Introduction

Fluid Structure Interaction (FSI) occurs when fluid flow exerts forces and moments on solid structures, causing the structures to move/deform in such a way that it perturbs the initial flow. This type of interaction causes the deformation of an aircraft wing during flight, and the vibration of a civil engineering structure

ISSN 1823-5514

due to airflow. FSI is a good example of complex flow problem over moving bodies, found in engineering.

The prediction of this interaction using Computational Fluid Dynamic (CFD) is very challenging. This is mainly due to the requirement of generating different grid, at every time step, to adapt well with the moving geometry. It is, therefore, advantageous to use meshless CFD methods, like the well-known vortex method, as a suitable tool to perform flow analysis over moving/deforming boundaries. Vortex Methods is non-conservative numerical method that resolves the Navier-Stokes equation (NS) in terms of vorticity field. The velocity fields are obtained from the calculated vorticity field using the Biot-Savart formulation. It is well-known that the Biot-Savart calculation requires long computation time.

One variant of vortex method is the so-called Vortex-In-Cell method (VIC). This method has been used to analyze FSI [1]. In this hybrid method, the Biot-Savart calculation is replaced by resolving the vorticity on grid cells. The velocity is calculated by solving the Poisson's equation for the streamfunction on the grids. Then, properties on each cell are redistributed back to particles to perform the convection process. Although, the VIC solver can handle complex geometries, the scheme still requires grid generation.

Another type of vortex method is the Particle Strength Exchange method, which has been employed to simulate the flow around complex bodies [2]. The difference between PSE and VIC is that the Laplacian is replaced by an integral operator, which is meshfree. The similarity of both methods is that they utilize the interpolation function to remesh the fluid domain. Nevertheless, the interpolation scheme is known to produce numerical dissipation error.

Leonard [3] proposed a purely Lagrangian core spreading vortex method, in which the diffusion term of NS is modeled by increasing the core size at each time step. The method is fully meshfree and seems to be easy to perform. However, the continuously increasing particle's core causes the solver to track the particles with their average velocity, rather than their local velocity. Additionally, Greengard [4] has shown that the core spreading vortex method, which enlarges the core size only, is not convergent to resolve the full Navier-Stokes equation. To deal with this problem, Rossi [5] resurrected the core spreading method by imposing the splitting scheme to control the evolution of the core size. Accordingly, the evolution of core size is constrained with a certain threshold. When a particle's core is greater than the threshold value, the particle is split into children particles. By using the splitting scheme, flow simulation around complex bodies can be conducted. However, the number of particles increases out of control. Therefore, the number of particles introduced in flow field is increasingly more than enough to represent the statistics of the solutions. Consequently, the computer memory is overflowed. To cope with this issue, the merging scheme is developed by the same author Rossi to reduce the memory, and to maintain the overlapping among particles. In particular, similar and nearby particles are merged into one satisfying the zero, the first, and the second multipole moments.

In the present work, the fast mesh-free solver, based on the core spreading vortex method, is developed to simulate flow over a moving body. The Boundary Element Method and Nascent vortex elements are introduced to enforce the no-through boundary condition for general geometry and no-slip boundary conditions for dynamic motions, respectively. The Fast Multipole Method is employed in order to accelerate the velocity computation. The splitting and merging spatial adaptation scheme is utilized to resurrect the core spreading method. Finally, the solver is validated by performing simulations of two external flow problems: flow around an impulsively started cylinder at Re = 550 and a forced-oscillating flat plate at Re = 10000.

Fast Lagrangian Vortex Method

The vortex methods are based on the momentuum equation and the continuity equation for incompressible flow which are written in vector form as follows:

$$\frac{\partial \underline{u}}{\partial t}+(\underline{u}\cdot\nabla)\underline{u}=-\frac{1}{p}\nabla p+\nu\nabla^2\underline{u} \tag{1}$$

$$\nabla.\underline{u}=0 \tag{2}$$

Taking the *Curl* of both equations (1) and (2) it follows:

$$\frac{\partial \underline{\omega}}{\partial t}+(\underline{u}\cdot\nabla)\underline{\omega}=(\underline{\omega}.\nabla)\underline{u}+\nu\nabla^2\underline{\omega} \tag{3}$$

$$\nabla^2 p=-\rho\nabla.(\underline{u}\nabla\underline{u}) \tag{4}$$

where $\underline{u}$ is velocity vector, p the pressure, and ρ the density. The vorticity $\underline{\omega}$ is defined as:

$$\underline{\omega}=\nabla\times\underline{u} \tag{5}$$

The pressure p can be independently calculated by the Poisson equation (4) once needed. Lagrangian expression for the vorticity transport expressed in Eq. (3) is then given by:

$$\frac{d\underline{\omega}}{dt}=(\omega.\nabla)\underline{u}+\nu\nabla^2\underline{\omega} \tag{6}$$

When a two-dimensional flow is dealt with, the first stretching term of the right hand side in Eq. (6) disappears and so the two-dimensional vorticity transport equation is simply reduced as diffusion equation:

$$\frac{d\underline{\omega}}{dt} = \nu \nabla^2 \underline{\omega} \tag{7}$$

In order to solve this equation numerically there is a need to approve by means of a viscous splitting algorithm. The algorithm includes two steps. The first step, the so-called convection, is to track particle elements containing the certain vortices with their own local convective velocity by Biot-Savart formulation:

$$\underline{u}(\underline{x},t) = \frac{1}{4\pi} \int \frac{\underline{\omega}(\underline{x}',t) \times (\underline{x} - \underline{x}')}{|\underline{x} - \underline{x}'|^3} d\underline{x}' \tag{8}$$

where $\underline{x}$ is vector of position. The term inside integral in Eq.(8) is integrated over all particles in the computational domain. The Biot-Savart relation is N-body problem that involves $O(N^2)$ evaluations. The calculation that involves $O(N^2)$ evaluations is called 'direct computation'. It makes this method not practical because of high memory requirement.

Fast multipole method

In order to overcome the N-body problem mentioned above, the Fast Multipole Method (FMM) is employed in this work to accelerate the velocity computation [6]. The method reduces significantly the velocity computation time due to the fact that interactions among particles are not computed directly. In more details, the FMM, first, constructs the data of particles by tree structure of box in which particles are laid on. Second, the direct interactions of box's centers are evaluated by using multipole expansions of all these centers. Finally, the interaction of all direct particle pairs is translated from these centers to their own particles. Therefore, it reduces amount of computation process to the order of $O(N)$. Reducing amount of computation process affects computational speed that is major problem in analyzing FSI.

Figure 1 shows the computational acceleration achieved by using FMM. In the figure, computational time using the direct Bio-Savart (Equation (8)) and FMM is plotted against the number of particle used during a simulation. As can be seen, the difference of computational time between the two methods is small up to around 200000 particles. However, the FMM acceleration increases significantly as the number of particle increases beyond 500000. Hence, the use of FMM allows for longer simulation time, since, in the developed vortex method, the number of particle increases with simulation time.

Boundary element method to satisfy the no-through boundary condition

Bounded flow problems require the enforcement of the no-through condition on boundaries. Vortex element method is a meshfree approach. Therefore, the enforcement of no-through boundary conditions is accomplished through the

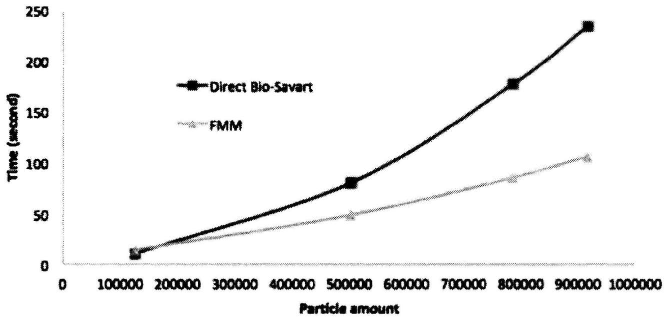

Figure 1: Comparison between Direct Bio-Savart and FMM computational time vs. number of particles

use of boundary element methods (BEM) [7]. The BEM calculates a vortex sheet's strength, which represents the slip velocity on the boundary necessary to satisfy no-through condition. In BEM, the boundary is discretized into panels and the vortex strength of each panel, Γ, is calculated. These vortex strengths or wall circulations represent initial vorticity vectors on the wall panels. The calculated vortex strength is a vector with two wall-tangent components and a normal component which satisfied the no-through condition.

Introduction of nascent vortex element to satisfy the no-slip boundary condition

In viscous flows, the no-slip and no-through boundary conditions on solid surface must be satisfied. Due to the introduction of Nascent vortex element [8], the no-through and no-slip boundary conditions are already satisfied. Figure 2 shows the sketch of the production of a Nascent Vortex Element.

In Figure 2, s_i, h_i, u_i denote respectively length of an outer boundary element, vorticity layer thickness and tangential velocity at each node of the outer boundary. The sketch in the figure is used to show the process of satisfying the wall boundary conditions by diffusing vortex elements from the wall. The Nascent vortex element is convected and diffused by velocities: V_c and V_d, respectively, as follows:

$$V_c = \frac{1}{s_i}\left(\frac{h_i u_i}{2} - \frac{h_{i+1} u_{i+1}}{2}\right) \tag{9}$$

$$V_d = \frac{dr_{diffusion}}{dt} = \frac{1.136^2 \nu}{r_{diffusion}} \tag{10}$$

where the height of boundary layer at certain panel i, h_i, is given by:

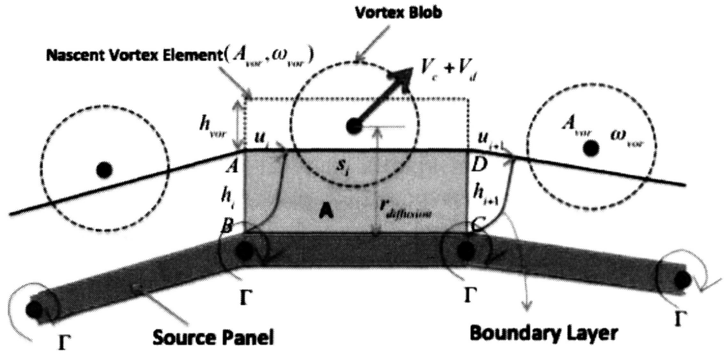

Figure 2: Production of Nascent Vortex Elements

$$h_i = r_{diffusion} = 1.136\sqrt{v\Delta t} \tag{11}$$

Also, the Nascent Vortex Element is replaced by an equivalent vortex blob with an area A and vorticity ω_{vor} as given by the following:

$$h_{vor} = \left(V_c + V_d\right)\Delta t \tag{12}$$

$$A_{vor} = h_{vor} \times s_i, A = h_i \times s_i \tag{13}$$

$$\omega_{vor} = \frac{\Gamma}{A + A_{vor}} \tag{14}$$

In Equation (14), Γ is the circulation originally involved in the element of vorticity layer [ABCD]. Actually, that is equal to the strength of vortex sheet calculated already by employing BEM, which satisfies no through boundary condition. Accordingly, the core size of the initial generated blob is also calculated by the formula

$$sig_{blob} = 2\sqrt{\frac{\Gamma}{\pi\omega_{vor}}} \tag{15}$$

Once a Nascent vortex element is shed from wall, new vortex element, which is satisfied the no-slip boundary condition above, is redistributed along the wall panel for the next time step.

Core Spreading Diffusion Model

In core spreading method, the core size magnitude is given by:

$$\sigma(t) = \sqrt{4v\Delta t} \tag{16}$$

which exactly represents for viscous diffusion. σ is the core radius of the vortex blob, and represents for the physical length scale of the vortex element. The rate of change of the core radius is:

$$\frac{d\sigma_i}{dt} = \frac{2v}{\sigma_i} \tag{17}$$

Equation (17) satisfies the diffusion equation (7). However, the total numerical truncation error, the so-called Lagrangian effect [4], increases proportional to the spreading rate of change of particle core size. Increasing core size of each particle i makes the particle advect with its average velocity rather than its local velocity [9]. Hence, there is a need of spatial adaptation to control core size of particle to be small enough to minimize the Lagrangian effect and maintain the spatial resolution.

Barba [9] proposed a method, which based on the Radial Basis Function Interpolation to redistribute the vorticity strengths field, using the smaller core sizes. The method leads to the linear system $\varsigma_{ij}\Gamma_i = \omega_i$, where ς_{ij} is Gaussian function of two vortex elements i and j, ω_i is the vorticity of the element i evaluated by heat kernel function. One of the advantages of this method is that the number of particles remains constant. And the overlapping among vortex elements is spatially adapted by reducing the core sizes into sufficiently small core sizes.

In this paper, we use the work proposed by Rossi [5], the so-called a splitting scheme, to spatially adapt the flow field. In particular, if the core radius of the vortex blob is larger than a threshold, then the "parent" blob is split into the several smaller "children" blobs, and the vortex strength of the parent are divided by the number of the children. The children core radius is reset into the smaller core radius. Obviously, the children cores are overlapped. Otherwise, the outstanding issue of the splitting scheme is to introduce the large amount of vortex elements. In other words, the number of vortex elements is introduced larger than the required vortex elements to sufficiently resolve the flow. Thus, the merging scheme is also proposed for the particle population control and for the overlapping control. The detail of two schemes is mentioned in the following sections.

Splitting scheme

The splitting scheme is proved convergent if a threshold core size σ_{max} is given to control evolution of core size over time [5]. On the other hand, as long as core size σ_j is larger than the threshold σ_{max}, the particle with core size σ_j would be split into a set of thinner core size particles where each particle inside the set has core size equal to $\alpha\sigma_j$. This set has also to satisfy the zero, the first moments of vorticity as follow:

$$\Gamma_p = \sum_{c=1}^{M} \Gamma_c \tag{18}$$

$$\Gamma_p \underline{x}_p = \sum_{c=1}^{M} \Gamma_c \underline{x}_c \tag{19}$$

where Γ_p, Γ_c stand for vorticity strengths of parent particles (before splitting), and children (after splitting) particles. M is the number of child particles. M is observed to be equal to 5 during our simulation for the good results. It is depicted as follow:

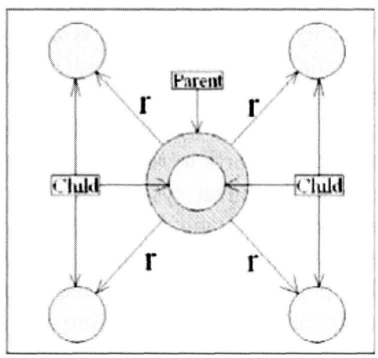

Figure 3: Spitting scheme around a parent particle

Where the free parameter r is given by:

$$r = \sigma_j \sqrt{2(1-\sigma^2)} \tag{20}$$

where α is overlapping parameter and set to be equal to 0.85.

Merging scheme

The subsequent number of numerical particles increases proportional to computational time. The splitting scheme does not allow for the long time simulation although it resolves the flow correctly in high spatial resolution. Also, more particles require more memory storage. In order to deal with this circumstance, Huang *et al.* [10] developed a merging spatial adaptation scheme. Accordingly, nearby particles are merged into one as shown in Figure 4.

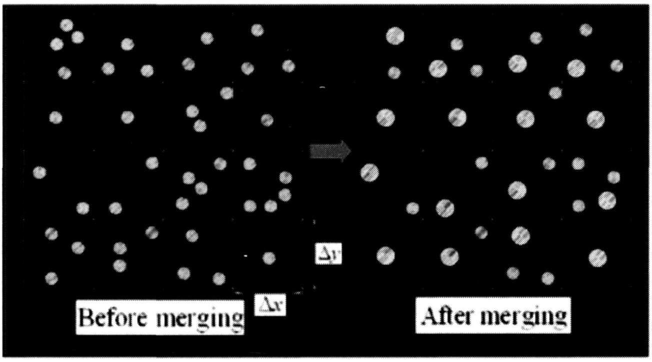

Figure 4: Illustration of merging scheme in Lagrangian Vortex Method

If (x_j, Γ_j, σ_j, $j = 1, \ldots N$) are the set of nearby particles, then those nearby particles are going to be replaced by one $\underline{x}_0$, Γ_0, σ_0 such that:

$$\Gamma_j = \sum_{j=1}^{N} \Gamma_j \tag{21}$$

$$\Gamma_0 \underline{x}_0 = \sum_{j=1}^{N} \Gamma_j \underline{x}_j \tag{22}$$

$$\Gamma_0 \sigma_0 = \sum_{j=1}^{N} \Gamma_0 \left(\sigma_0^2 + \left|\underline{x}_0 - \underline{x}_j\right|^2\right) \tag{23}$$

Meanwhile following thresholds should be satisfied:

$$\Gamma_0 < \Gamma_{ref} \varepsilon \alpha^2 \alpha_{max}^2 \tag{24}$$

$$\sigma_0 < \sigma_{max} \tag{25}$$

where Γ_{ref} and ε are the reference vorticity strength, and the error tolerance, respectively. In the case of impulsively started cylinder at Re = 550, ε is set to be equal to 1, and $\Gamma_{ref} = UD$.

Simulations

Impulsively started cylinder at Re = 550

In order to validate the performance of the vortex method solver, we performed the simulations of flow over an impulsively started circular cylinder at Reynolds number 550. The detail simulation parameters are listed in Table 1.

Table 1: Input parameters

Re	Δt (time step)	Panels
550	0.01(s)	200

In this solver, the Reynolds number is the one of the primary input parameters for simulation. Another major parameter is the time step, which affects the accuracy of the solver in convection step, as well as to control the spatial adaptation error in merging event. Large time step Δt increases the number of merging events in the same time period of simulation. Accordingly, the error is spatially integrated within the simulation time. Additionally, the tolerance parameter ε also plays an important role to constrain the existence of merging events. If the tolerance parameter is set to be large, the frequency of the merging events is high and vice versa. In order to control the particle's population, parameter Γ_{trim} is introduced. The smaller the value of Γ_{trim}, the larger memory requirement. Based on a number of tests, we set $\Gamma_{trim} = \nu \times 10^{-4}$, where ν is kinematic viscosity of the flow. The last parameter, which is not the least important, is number of panels. The number of panels determines the initial vorticity surrounding the wall through the introduction of Nascent vortex element, as mentioned in Section 3. This vorticity layer apparently represents for the no-though boundary condition, which is the input for the no-slip boundary condition to introduce the Nascent element, as shown in Section 3. The accuracy and the stability of the vortex method solver obviously depend on this initial condition. In addition, moving boundary conditions can be enforced during this process in order to perform more complex viscous flow simulations [7].

The results of the simulation, which shows the vortex shedding process behind the cylinder, are depicted in Figure 5. The left hand side is the current results, and the right hand side is the same simulation conducted by Ploumhans and Winckelmans [2]. The figure shows small differences, between the current and the reference results, in the shedding patterns behind the cylinder. However, the location of the stagnation points are found to be at the same location as in the reference. Hence, it can be concluded that the viscous diffusion model effects the shedding structures considerably, while it has minor influence on boundary layer.

Figure 6 shows the positions of vortex bubble's center at different simulation times (T = 1 (s), T = 3 (s), and T = 5 (s)). As depicted in the figure, initially (T = 1), there is a notable deviation between the position obtained in the present simulation and that of the reference. However, this deviation becomes smaller and smaller as time progresses. In fact, at later times (T = 5 (s)) the position of bubble's center calculated using the present method approaches that

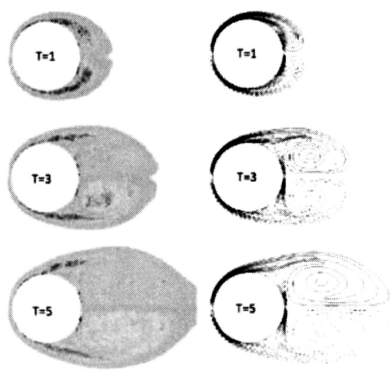

Figure 5: Comparison of the calculated vorticity contours with reference. Left hand sight: present simulation. Right hand sight: Ploumhans and Winckelmans [2]

of the reference. The initial difference is probably due to the fact that different method is used to enforce the no-slip boundary conditions.

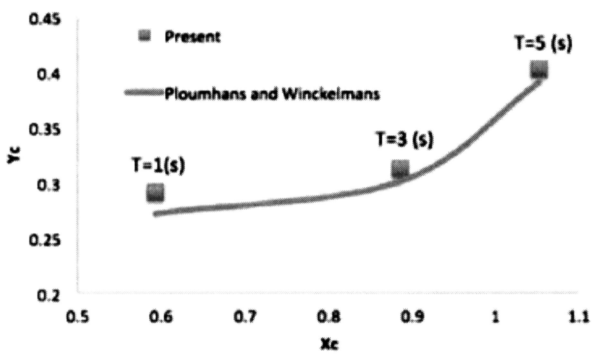

Figure 6: Comparison of the calculated position of vortex bubble's center. Present (dashed line), Ploumhans and Winckelmans (contiuous line)

Figure 7 shows the time history of drag coefficients. The method to caltulate drag coefficient (Cd) is the linear impulse method, which is described in [2]. During the early stage of simulation, there is a slight deviation between the present drag coefficients and that obtained by Ploumhans and Winckelmans [2]. This is due to the fact that there are still many unnecessary splitting and merging events in this early stage of simulation. This is also the another probable source of error causing the difference in the earlier time vortex bubble's center, discussed

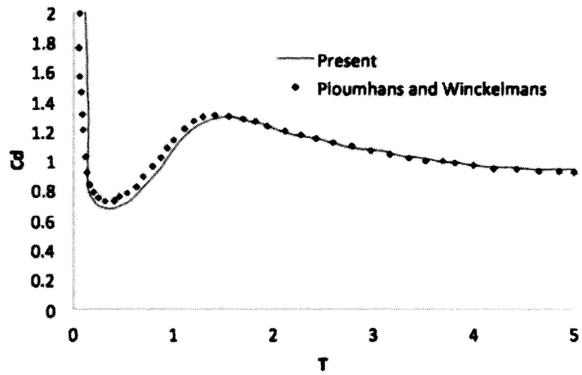

Figure 7: Drag coefficient. Dashed line is current result, continuous line is result of Ploumhans and Winckelmans

in previous paragraph. However, as the simulation progresses, the present calculated Cd values are in very good agreement with that of the reference.

Flutter speed of forced-vibrating flat plate at Re = 10000

In engineering simulations, it is sometimes difficult to satisfy the exact boundary conditions, due to the combination of complex geometry and complex motion of the boundary. Here, the developed method has advantages over the more conventional methods in dealing with such problems.

In order to show the capability of the developed solver for analyzing FSI problems, forced-translation and forced-rotation simulations for flat plate are performed using initial conditions listed in Table 2. The forced motion of the plate is defined as sinusoidal function for both translational ($y = 0.25\cos(\omega t)$) and rotational ($\theta = 0.3\cos(\omega t)$) modes. The simulation is performed in a limited range of frequency of the vertical and rotational motions ω = 0.5, 1, 2, 3. The configuration of the simulation is depicted in Figure 8.

Table 2: Initial parameters

Re	t/B (Thickness ratio)	Panels
10000	0.1	500

Accordingly, the current solver is then used to determine the unsteady aerodynamic forces [2] and moments [11] for every time step. The loads, displacement, and rate of displacement data are recorded, and used for

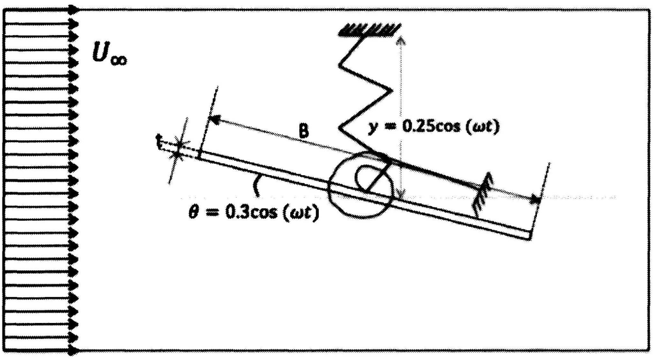

Figure 8: Configuration of the forced-vibrating flat plate simulation

calculations of flutter derivatives. Based on these calculated flutter derivatives, the flutter speed of the flat plate can be predicted. The details of the method used to determine the flutter derivatives is the same as that described in [12].

Figure 9 depicts the calculated flutter derivatives (H1*-H4* and A1*-A4*) versus the reduced velocity. The figure shows that the current results are in good agreement with Theodorsen's analytical solution [7] for smaller values of reduced velocities. On the other hand, the results tend to deviate from the analytical solutions for higher reduced velocities. However, it is important to remember that the analytical solution is obtained by imposing the linearity assumption.

The flutter speed calculation result is shown in the Figure 10, which depicts the plot of imaginary and real part ratio (K_i/K_α) of reduced frequency versus reduced velocity of torsional mode, U_θ. The continuous line is interpolated by using the least square method based on the dashed line data. From the plot, it is estimated that the value of K_i/K_α equals to zero when U_θ is approximately equal to 6.46 (m/s).

Robertson [13] analyzed the same case also using different numerical method (conventional mesh-based CFD). Table 3 shows the comparison between the current result and that found in reference [13]:

The flutter speed calculated using the current solver differs about 4% from the reference. The relatively small 4% difference is probably caused by the difference in the exact geometry and detail of simulation parameters used in the two flutter speed calculations.

Conclusions

In conclusion, we have developed an accelerated two-dimensional core spreading vortex method for high resolution flow simulation. The computational time is

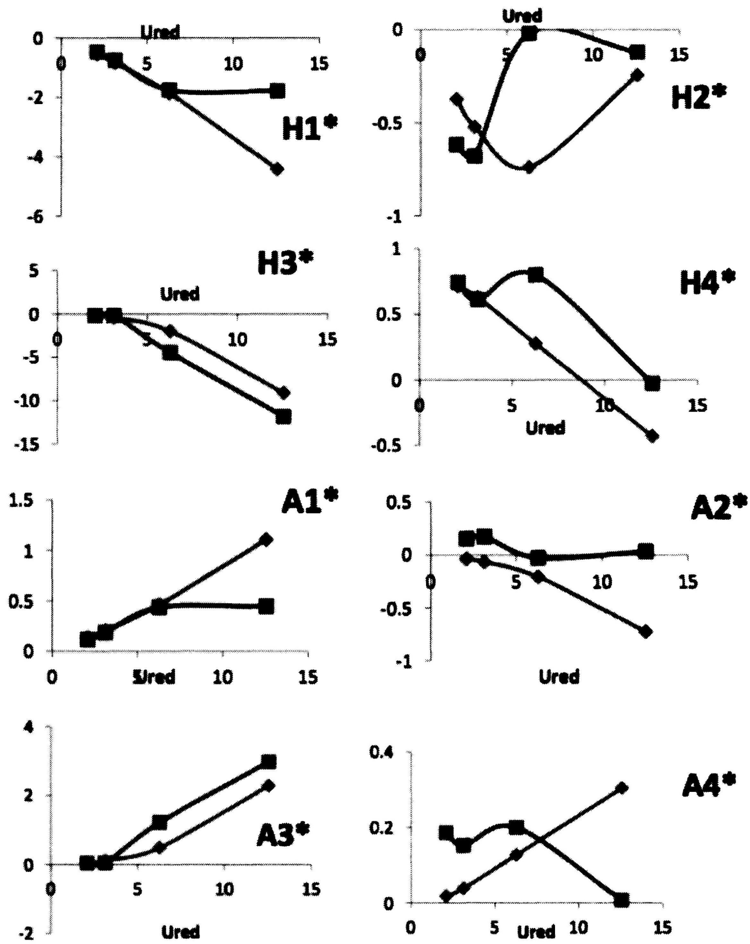

Figure 9: Flutter derivatives versus reduced velocity. Red line is current results, blue lines is Theodorsen's analytical result

accelerated by using the FMM. The spatial resolution is maintained during the simulation by implementing the splitting and merging scheme. It has been demonstrated that the developed meshfree CFD method can accurately simulate flows over moving boundary. The solver is first validated by performing the simulation of flow around an impulsively accelerated cylinder at Reynolds number 550. The results are found to be in good agreement with those reported in literature. In order to show the capability of the solver in simulating moving boundary problem and investigate the possibility of using the method to conduct FSI analysis, we performed the simulation of flow over a forced-vibrating flat

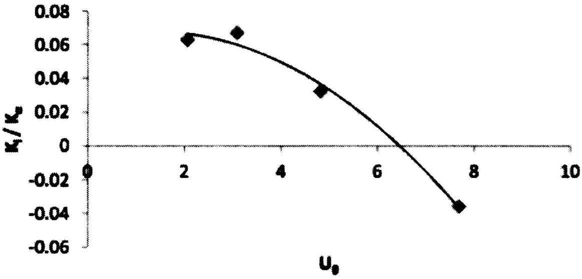

Figure 10: Plot of K_i/K_a versus U_θ

plate at Reynolds number 10000. The results of the simulation are then used to estimate the flutter speed of the plate. The results are found to be consistent with analytical solutions and available reference.

Table 3: Error analysis

	Present	Robertson	error
U_θ	6.46 (m/s)	6.21 (m/s)	4%

Acknowledgement

The authors would like to gratefully thank AUN/SEED-Net project No. ITB CR1301 for supporting the current research works. This research was also partly supported by a research grant under the administration of Bandung Institute of Technology, Indonesia. We thank Eky V. F. for various discussions that contributed to the quality of this work.

References

[1] G.-H. Cottet and P. Poncet. (2004). "Advances in direct numerical simulations of 3D wall-bounded flows by vortex-in-cell methods", J. Computational Physics 193 (1), 136-158.

[2] P. Ploumhans and G.S. Winckelmans. (2000). "vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry", J. Computational Physics 165 (2), 354-406.

[3] A. Leonard. (1980). "Vortex methods for flow simulations", J. Computational Physics 37, 289-335.

[4] C. Greengard. (1985). "The core-spreading vortex method approximations the wrong equation", J. Computational Physics 61, 345-348.
[5] L. Rossi. (1996). "Resurrecting core-spreading vortex methods: a new scheme that is both deterministic and convergent", SIAM Journal on Scientific Computing 17(2), 370-397.
[6] L. Greengard and V. Rokhlin. (1978). "A fast algorithm for particle simulations", J. Computational Physics 73, 325-348.
[7] G. Morgenthal. (2002). "Aerodynamic analysis of structures using high-resolution vortex particle methods", Dissertation, University of Cambridge.
[8] K. Kamemoto. (2005). "On contribution of advanced vortex element methods toward virtual reality of unsteady vortical flows in the new generation of CFD", Brazilian Congress of Thermal Sciences and Engineering 26(4), 1678-5878.
[9] L.A. Barba, A. Leonard and C. B. Allen. (2005). "Advances in viscous vortex methods-Meshless spatial adaption based on radial basis function interpolation", *Int. J. Numerical Methods in Fluids* 47(5), 387-421.
[10] M.J. Huang, H.X. Su and L.C. Chen. (2008). "A fast resurrected core-spreading vortex method with no-slip boundary conditions", J. Computational Physics 228(6), 1916-1931.
[11] F. Noca, D. Shiels and D. Jeon. (1997). "Measuring instantaneous fluid dynamics forces on bodies, using only velocity fields and their derivatives", J. Fluids and Structures 11(3), 345-350.
[12] L.R. Zuhal, E.V. Febrianto and D.V. Duong. (2014). "Flutter speed determination of two degree of freedom model using discrete vortex method", Applied Mechanics and Materials 660, 639-643.
[13] I. Robertson, S.J. Sherwin and P.W. Bearman. (2003). "Flutter instability prediction techniques for bridge deck sections", Int. J. For Numerical Methods in Fluids 43, 1239-1256.

JOURNAL OF MECHANICAL ENGINEERING (JMechE)

Aims & Scope

Journal of Mechanical Engineering (formerly known as Journal of Faculty of Mechanical Engineering) or JMechE, is an international journal which provides a forum for researchers and academicians worldwide to publish the research findings and the educational methods they are engaged in. This Journal acts as a vital link for the mechanical engineering community for rapid dissemination of their academic pursuits.

Contributions are invited from various disciplines that are allied to mechanical engineering. The contributions should be based on original research works. All papers submitted to JMechE are subjected to a reviewing process through a worldwide network of specialized and competent referees. To be considered for publication, each paper should have at least two positive referee's assessments.

General Instructions

Manuscripts intended for publication in JMechE should be written in camera ready form with production-quality figures and done electronically in Microsoft Word 2000 (or above) and submitted online to jmeche.int@gmail.com. Manuscripts should follow the JMechE template.

All papers must be submitted online to **jmeche.int@gmail.com**

Correspondence Address:

Editor In Chief
Journal of Mechanical Engineering (JMechE)
Faculty of Mechanical Engineering
Universiti Teknologi MARA
40450 Shah Alam, Malaysia.

Tel : 603 – 5543 6459
Fax : 603 – 5543 5160
Email: jmeche.int@gmail.com
Website: http://fkm.uitm.edu.my/jmeche

Paper Title in Arial 18-point, Bold and Centered

Author-1
Author-2
Affiliation of author(s) from the first institution

Author-3
Affiliation of the author(s) from the second institution in Times New Roman 10 italic

ABSTRACT

The first section of the manuscript should be an abstract, where the aims, scope and conclusions of the papers are shortly outlined, normally between 200 and 300 words. TNR-10 italic

Keywords: *maximum 5 keywords.*

Title of First Section (Arial 11 Bold)

Leave one blank line between the heading and the first line of the text. No indent on the first para after the title; 10 mm indent for the subsequent para. At the end of the section, leave two blank lines before the next section heading. The text should be right and left justified. The recommended font is Times New Roman, 10 points. In 152 mm x 277 mm paper size, the margins are: left and upper: 22 mm each; right: 20 mm, lower: 25 mm.

Secondary headings (Arial 10 Bold)

The text starts in the immediately following line. Leave one blank line before each secondary heading.

Tertiary headings (Arial 10)

If they are required, the tertiary headings shall be underlined. Leave one blank line before tertiary headings. Please, do not use more than three levels of headings, try to keep a simple scheme.

Tables and illustrations should be numbered with arabic numbers. Tables and illustrations should be centred with illustration numbers written one blank

Tables and illustrations should be numbered with arabic numbers. Tables and illustrations should be centred with illustration numbers written one blank line, centered, after the relevant illustration. Table number written one line, centered, before the relevant table. Leave two blank lines before the table or illustration. Beware that the proceedings will be printed in black and white. Make sure that the interpretation of graphs does not depend on colour. In the text, tables and figures should be referred to as Figure 1 and Table 1.

The International System of Units (SI) is to be used; other units can be used only after SI indications, and should be added in parenthesis.

Equations should be typed and all symbols should be explained within the manuscript. An equation should be proceeded and followed by one blank line, and should be referred to, in the text, in the form Equation (1).

$$y = A + Bx + Cx^2 \tag{1}$$

Last point: the references. In the text, the references should be a number within square brackets, e.g. [3], or [4]–[6] or [2, 3]. The references should be listed in numerical order at the end of the paper.

Journal references should include all the surnames of authors and their initials, year of publication in parenthesis, full paper title within quotes, full or abbreviated title of the journal, volume number, issue number and pages.

Examples below show the format for references including books and proceedings

Examples of references:

[1] M. K. Ghosh and A. Nagraj, "Turbulence flow in bearings," Proceedings of the Institution of Mechanical Engineers 218 (1), 61-4 (2004).

[2] H. Coelho and L. M. Pereira, "Automated reasoning in geometry theorem proving with Prolog," J. Automated Reasoning 2 (3), 329-390 (1986).

[3] P. N. Rao, Manufacturing Technology Foundry, Forming and Welding, 2nd ed. (McGraw Hill, Singapore, 2000), pp. 53-68.

[4] Hutchinson, F. David and M. Ahmed, U.S. Patent No. 6,912, 127 (28 June 2005).