FEDSM2005-77024

A LAGRANGIAN VORTEX METHOD FOR SIMULATING FLOW OVER 3-D OBJECTS

A. Gharakhani, Sc.D. Applied Scientific Research Santa Ana, CA adrin@Applied-Scientific.com J. Sitaraman, Ph.D. University of Maryland College Park, MD M. J. Stock Applied Scientific Research Santa Ana, CA

ABSTRACT

A grid-free method for the simulation of incompressible flow over complex 3-D objects is presented. The algorithm is based on the Lagrangian Vortex Element and the Vorticity Redistribution methods to account for vorticity dynamics and diffusion, respectively. The wall velocity boundary conditions are imposed by solving a Petrov-Galerkin discretization of the Fredholm boundary integral equation of the second kind for the vector-valued vortex sheets at the wall. Computations are accelerated using a MPI-parallel adaptive Fast Multipole Method (oct-tree code). In this paper, the computational algorithm is described briefly, and the first in a series of benchmark results presented using the example of flow over a sphere at low Reynolds numbers. Preliminary diagnostics of the present simulations show very good agreement with previously published data.

Keywords: Grid-free CFD, vortex methods, sphere.

INTRODUCTION

Lagrangian Vortex-Boundary Element Methods (LVBEM) are a class of particle-based computational techniques best suited for simulating vortex-dominated incompressible flow in/around complex geometries. They are *grid-free* in the fluid domain; thus, reducing problem setup time significantly. Furthermore, they are ideal for vortex-dominated flow simulations thanks to the minimal numerical diffusion and the dynamic adaptivity of the algorithm. Additionally, continuity and far-field boundary conditions are satisfied by construction.

The first 3-D LVBEM (for internal flows) was developed by Gharakhani & Ghoniem and applied to examples of interest to industry [1-5]. However, these simulations used the random walk method for diffusion, which is known for its noise, relatively low accuracy and convergence rate. An advanced LVBEM was later introduced by Ploumhans *et al.* [6], and its robustness as a Direct Numerical Simulation (DNS) tool demonstrated using the example of flow over a sphere at three Reynolds numbers up to 1,000. The details of the latter are beyond the scope of this paper, the primary relevant issue being the use of Particle Strength Exchange (PSE) [7] for predicting diffusion. PSE has been used by a number of research groups; however, it suffers from a few disadvantages. First, it is guaranteed to conserve *only* the zeroth moment of the diffusion equation, but not the higher moments. Second, the PSE formulation requires the explicit usage of the elemental control volume, which is not clearly defined and/or easily obtained in particle-based methods. Indeed, it is generally meaningless to talk about control volumes in the context of grid-free computing. Third, with PSE it is necessary to "regrid" the particles onto a uniform distribution once every few timesteps to maintain its accuracy. This diminishes the appeal of vortex methods as a "grid-free" tool with minimal numerical diffusion, since regridding is associated with certain complications for complex geometries, and the application of traditional projection schemes [6] introduces numerical errors back to the computations. A newly proposed projection scheme based on radial basis functions holds strong promise for maintaining accuracy, but it is presently quite costly [8].

The advanced algorithm described in this paper uses the Vorticity Redistribution Method [9,10] (VRM) for predicting diffusion, which provides a number of significant advantages. First, the conservation properties of the diffusion equation are preserved to arbitrarily high order [9,10]. Second, VRM does not require frequent remeshing of the particles onto uniform "grids" to maintain accuracy [11]. Third, VRM diffuses circulation, not vorticity; therefore, it does not need control volumes explicitly. Fourth, unlike other grid-free deterministic diffusion solvers, VRM is capable of diffusing singular (not smoothed) particles. This is quite significant because singular particles represent the *exact* discrete solution of the vorticity transport equations [12], and are thus ideal as DNS algorithms [11]. In contrast, smoothed elements discretize the convolution of the transport equations with the smoothing function and are, therefore, of Large Eddy Simulation (LES) character [12]. Another significant advantage of VRM over other grid-free methods is its ability to detect "holes" within the neighborhood of a diffusing element, in which case a judicious

filling of the holes with new elements leads to a guaranteed stable solution. "Holes" are usually developed when severe stretching of the vorticity field disperses the vortex elements, and/or when the vorticity field front expands due to diffusion [13].

In what follows, the algorithm for the present LVBEM is described briefly. The accuracy of the latter is then demonstrated using the benchmark problem of impulsively started flow over a sphere at low Reynolds numbers.

NOMENCLATURE

- A_i area of panel *i*
- N_P number of triangular panels
- N_V number of singular vortex particles
- N_n number of particles in the neighborhood of a diffusing particle
- *R* radius of a sphere, centered on a diffusing particle, where new particles are inserted due to diffusion
- Re Reynolds number
- Re_D Reynolds number based on the diameter D
- *T* non-dimensional period of a sinusoidal cycle
- \vec{U}_{∞} freestream velocity
- f_{pq} fraction of circulation given off from particle p to particle q due to diffusion
- \vec{n} unit normal vector pointing into the fluid domain
- t time
- \vec{u} velocity vector
- \vec{u}_{σ} smooth velocity vector, with smoothing core radius σ
- $\vec{u}_{\sigma,p}$ smooth velocity vector at particle p
- \vec{x} position vector
- x, y, z components of position vector \vec{x}
- \vec{x}_{pq} difference in position vectors of particles p and q, $\vec{x}_p - \vec{x}_q$

Greek letters

- $\vec{\nabla}$ gradient operator
- Δt computational timestep
- $\vec{\Gamma}_p$ vectorial circulation of particle p
- Ω computational fluid domain
- $\partial \Omega$ bounding surface of fluid domain
- $\delta(.)$ Dirac Delta function
- $\vec{\gamma}$ vortex sheet strength vector
- $\vec{\omega}$ vorticity vector
- σ core radius of smoothing function

Subscripts

p,q index to particles p and q, respectively

FORMULATION

The 3-D vorticity transport equations are [2,6]:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{u} \cdot \vec{\nabla} \vec{\omega} = \vec{\omega} \cdot \vec{\nabla} \vec{u} + \frac{1}{\text{Re}} \nabla^2 \vec{\omega}$$
(1a)

 $\vec{\nabla} \cdot \vec{u} = 0 \tag{1b}$

where the proper initial and boundary conditions (for velocity) are applied to Eq. (1).

In the present algorithm vorticity is discretized using N_V singular vortex particles, each with "vectorial circulation" $\vec{\Gamma}_n$:

$$\vec{\omega}(\vec{x},t) = \sum_{p=1}^{N_v} \vec{\Gamma}_p(t) \delta(\vec{x} - \vec{x}_p)$$
⁽²⁾

The particle velocities (and their gradients) are smooth in these computations and are evaluated by convolving the Biot-Savart integral for velocities with a smoothing function:

$$\vec{u}_{\sigma}(\vec{x},t) = \sum_{p=1}^{N_{v}} K_{\sigma}(\vec{x}-\vec{x}_{p}) \times \vec{\Gamma}_{p}(t) + \vec{U}_{\infty}(\vec{x},t)$$
(3a)

$$K_{\sigma}(\vec{x}) = K(\vec{x}) \int_{0}^{|\vec{x}|/\sigma} g(r) r^{2} dr$$
 , $K(\vec{x}) = -\frac{\vec{x}}{4\pi |\vec{x}|^{3}}$ (3b)

where $g(r) = (3/4\pi) \exp(-r^3)$ is the smoothing/core function of choice in this paper. Note that the smooth velocity gradient is obtained by differentiating Eq. (3) directly [2].

Given the above discretizations for vorticity and velocity, the Lagrangian evolution of the vorticity field is evaluated via

$$\frac{d\bar{x}_p}{dt} = \bar{u}_{\sigma,p} \tag{4a}$$

$$\frac{d\bar{\Gamma}_p}{dt} = \bar{\Gamma}_p \cdot \bar{\nabla} \vec{u}_{\sigma,p} \qquad \qquad p = 1, \cdots, N_V \qquad (4b)$$

$$\frac{\partial \vec{\omega}_p}{\partial t} = \frac{1}{\text{Re}} \nabla^2 \vec{\omega}_p \tag{4c}$$

Note again that Eqs. (4a-c) represent the discrete solution of Eqs. (1a-b), provided the velocity field in the former is given by \vec{u}_{σ} . Equations (4a-c) represent a viscous splitting algorithm, whereby the convection and stretch of vorticity are evaluated along particle trajectories, followed by the solution of the diffusion equation in an Eulerian frame [14]. A secondorder predictor-corrector time integration is used for Eqs. (4ab).

The VRM solution of diffusion begins by assuming that each particle p gives off a fraction of its circulation to particle q in its neighborhood due to the diffusion process, such that

$$\vec{\omega}(\vec{x}_p, t + \Delta t) = \sum_{q=0}^{N_a} \vec{\Gamma}_p(t) f_{pq} \delta(\vec{x} - \vec{x}_q)$$
(5)

where q = 0 represents p itself. Minimizing the residuals of the zeroth through second moment integrals of Eq. (4c), performing an Euler time integration, and substituting (5) yields a system of 10 equations and N_n unknowns for the fractions [9,10,11]:

$$\sum_{q=0}^{N_{a}} f_{pq} = 1$$
 (6a)

$$\sum_{q=1}^{N_a} f_{pq} \vec{x}_{pq} = 0$$
 (6b-d)

$$\sum_{q=1}^{N_n} f_{pq} x_{pq} y_{pq} = \sum_{q=1}^{N_n} f_{pq} y_{pq} z_{pq} = \sum_{q=1}^{N_n} f_{pq} z_{pq} x_{pq} = 0$$
(6e-g)

$$\sum_{q=1}^{N_q} f_{pq} \vec{x}_{pq}^2 = 2\Delta t / \text{Re}$$
(6h-j)

The system of equations (6) is under-determined, which is solved in the L_{∞} -norm using linear programming [9,10,11]. If the equations do not have a solution, new particles are inserted on a sphere of radius $R\sqrt{\Delta t/\text{Re}}$ until a solution is obtained. Stability analysis shows that $R \ge \sqrt{6}$; in this paper we use $R = \sqrt{8}$. Note that this particle insertion and solution strategy ensures that the nominal inter-particle spacing is $O(R\sqrt{\Delta t/\text{Re}})$ at all times and throughout the computational domain. The radius of the sphere of influence containing the N_n neighboring particles; i.e., the stencil size, is set to $\sqrt{17\Delta t/\text{Re}}$. Note that the stability of the smooth core functions requires that the core radius σ be larger than the inter-particle spacing. In this paper, $\sigma = 1.5R\sqrt{\Delta t/\text{Re}}$ is used to guarantee stability.

The computational complexity of VRM is $O(N_v)$. On the other hand, the standard Biot-Savart evaluations are of $O(N_v^2)$ complexity, making the utility of 3-D vortex methods impractical for most problems. To this end, an adaptive oct-tree code with complexity $O(N_v \log N_v)$ has been implemented to speed up the computations [15]. Further speedup is achieved via efficient MPI-based parallelization of the velocity and VRM evaluations for a distributed computing environment using the standard Orthogonal Recursive Bisection (ORB) algorithm.

To satisfy the wall velocity boundary conditions, vortex sheets are generated at the wall with their surface-tangent values $\vec{\gamma}$ obtained subsequent to the solution of the following Fredholm boundary integral equation of the second kind:

$$\frac{1}{2}\vec{\gamma}(\vec{x}) \times \vec{n}(\vec{x}) + \int_{\partial\Omega} \vec{\gamma}(\vec{x}') \times K(\vec{x} - \vec{x}') d\vec{x}' = \vec{U}_s(\vec{x})$$
(7a)

$$\vec{U}_{s}(\vec{x}) = \vec{U}_{\infty}(\vec{x}) - \int_{\Omega} \vec{\omega}(\vec{x}') \times K(\vec{x} - \vec{x}') d\vec{x}'$$
(7b)

The solution of Eq. (7) is obtained by discretizing the surface of the object with N_p contiguous triangular panels and assuming a shape function for $\vec{\gamma}$. We have implemented a collocation-based solution assuming a piecewise constant distribution of $\vec{\gamma}$, as well as a Petrov-Galerkin solution assuming a singular particle distribution. The latter is presented in this paper. Note that in the Petrov-Galerkin formulation, Eq. (7) is integrated with respect to \vec{x} , yielding the following area-integrated boundary element system:

$$\frac{1}{2}\vec{\Gamma}_{i} \times \vec{n}_{i} + \sum_{j \neq i}^{N_{p}} \vec{\Gamma}_{j} \times \int_{S_{i}} K(\vec{x} - \vec{x}_{j}) d\vec{x} = \vec{Q}_{s} \qquad (8a)$$
$$i = 1, \cdots, N_{p}$$

$$\vec{Q}_s = \int_{S_i} \vec{U}_{\infty}(\vec{x}) d\vec{x} - \sum_{p=1}^{N_v} \vec{\Gamma}_p \times \int_{S_i} K(\vec{x} - \vec{x}_p) d\vec{x}$$
(8b)

where $\vec{\Gamma}_i = \vec{\gamma}_i A_i$ on the surface is the "panel circulation" and is equal to the particle circulation $\vec{\Gamma}_p$ that is subsequently diffused into the fluid domain. The surface integrals of the velocity kernel are evaluated analytically [16] to get accurate solutions in situations where the source and target points are close to each other. Note that in Eq. (8b) the area-integrated influence of the vortex particles is evaluated using the singular, and not the smooth, velocity kernel to allow analytic integration. Also note that Eq. (8) is easily extendible to the case where the object experiences motion, including rotation, which will be presented in a future publication.

The system of Eqs. (8) is densely populated and of $O(N_P^2)$ in terms of both storage and CPU requirements. To this end, the adaptive oct-tree code for the particles is tailored to the case with panels to reduce the memory and CPU cost of Eqs. (8) to $O(N_P \log N_P)$, which is then solved iteratively via GMRES.

The link between the vorticity generated at the wall and the interior domain is accounted for via diffusion with either a Neumann or a Dirichlet boundary condition. We have experimented with the Dirichlet boundary condition as well as the Neumann boundary condition used in [6,17] for flow over a sphere and have found very little difference in the results. The VRM formulation for the problem of wall-bounded diffusion in a semi-infinite domain, conserving zeroth through second moments, has been derived by our group. According to this formulation, the vortex sheets retain a fraction of their strength at the wall and give off the remainder to two "layers of particles" adjacent to the wall, such that the moment conditions are preserved. Note that since VRM can be used to diffuse singular particles and sheets, difficulties encountered by other methods, such as "loss of vorticity/circulation" due to the core overlap of smooth vortex elements with the wall, are absent. In this paper, a simple strategy is used, which conserves the circulation and the first moment of diffusion from a wall with Dirichlet boundary condition. To this end, at each timestep, prior to solving Eq. (8) to satisfy the wall velocity boundary condition, all vortex particles within a numerical boundary layer of thickness $R_{\sqrt{\Delta t/\text{Re}}}$ are removed. Then, having solved Eq. (8), the circulation of each triangular panel is given all off to a corresponding new particle, which is positioned in the fluid along the normal passing through the panel centroid at an elevation of $h = 2\sqrt{\Delta t/(\pi \operatorname{Re})}$. This approach stabilizes the simulation and inhibits the unnecessary growth of the number of vortex elements that would otherwise be required if new vortex particles were "injected" into the flow at every timestep.

The proposed LVBEM for each timestep is:

(1) For a given distribution of vortex particles, evaluate the velocity field and its gradients via the adaptive FMM implementation of Eq. (3).

- (2) Convect and stretch the vortex particles according to (4a) and (4b), respectively.
- (3) Repeat steps (1) and (2) to complete a second-order time integration.
- (4) Diffuse the vorticity field (modify particle circulation) via VRM formulation, Eq. (6).
- (5) Remove the particles within the numerical boundary layer of thickness $R\sqrt{\Delta t/\text{Re}}$, as well as elements that might have jumped out of the fluid domain.
- (6) Impose the wall velocity boundary conditions by solving Eq. (8) for the corresponding wall vortex sheet vectors, using adaptive FMM.
- (7) Insert new particles in the numerical boundary layer, $h = 2\sqrt{\Delta t/(\pi \operatorname{Re})}$ away from the centroid of each vortex panel.

RESULTS

The salient features of the proposed methodology are presented in this section. To this end, we begin by showing our FMM speedups, defined as the ratio of the $O(N^2)$ CPU time to that of FMM, for both vortex particles and panels. Figure 1a depicts a comparison of our particle FMM speedups with the speedup results by Strickland et al. (SE) [18] and Warren & Salmon (WS) [19]. The test problem used - smooth vortex particles distributed randomly in a cube with RMSerror $\approx 10^{-4}$ - mimics that of SE as closely as possible. The result by WS is for particles distributed on a spherical shell and is used for qualitative comparison only. First thing to note is that our FMM timing breaks even with the $O(N^2)$ timing (unity speedup) at roughly 500 particles. In contrast, the breakeven points for SE and WS are at 70,000 and 20,000 particles, respectively. At 70,000 particles our FMM already provides a nine-fold speedup. At 10⁶ particles the speedup by our FMM is 85, which is roughly 14 and 4 times higher than the speedup reported by SE and WS, respectively. It should be added here that the SE and WS computations involve particle velocities only, whereas our computations include velocity gradients as well, which increases the CPU time by about a third. Also note that our FMM evaluations seem to asymptote to $O(N^{1.1})$ beyond 600,000 particles. Figure 1b depicts a comparison of our FMM speedup with that of Cheng et al. (GR) [20], who are the leaders in the field. For this case, GR's "black box" code was used on our computer for a more direct comparison. The test used randomly distributed smooth vortex elements in a cube. The RMS-error by our FMM was 3×10^{-4} , and it was 6×10^{-4} by the GR method. The breakeven point was fewer than 500 particles for our FMM, and 5,000 particles for GR's. In general, it can be claimed that our FMM is as robust as that of GR. Note that the observed differences between our results in Figs. 1a and 1b are due only to differences in RMS-errors.

Figure 2 depicts the speedups obtained by our FMM-BEM for a sphere and a cube. The open circles imply that the CPU times for the classical/direct method were obtained by extrapolation. The breakeven point for the two cases tested was roughly 200 panels, and depending on the geometry, speedups of O(100) can be achieved for as few as 30,000 panels. It is interesting to note that the largest number of panels one can use



Figure 1: FMM speedups by various authors

to solve the $(2N_p \times 2N_p)$ BEM problem by the classical method is 5,590 on a computer with 1 GB of memory. In contrast, we have successfully used FMM for up to 40,000 panels. We add here that we have implemented and tested Gauss-Seidel, Jacobi and GMRES for iterating the matrix solution to convergence, and we have found the CPU price-performance of each to depend on the geometry and flow conditions. For example, for the case of a cube, Gauss-Seidel (and, equally, Jacobi with 0.8 under-relaxation factor) achieves convergence roughly 30% *faster* than GMRES. However, Gauss-Seidel becomes 25% *slower* than GMRES for ovoids with high aspect ratio or other more complicated geometries. We are currently using GMRES as the default.

Figure 3 depicts preliminary results for the parallelization efficiency of the code on a cluster of four dual-processor AMD Opteron 246 computers, which are inter-connected with gigabit switches. Since the FMM component comprises over 90% of the simulation cost, the figure essentially depicts the parallel efficiency of the FMM. Efficiency is defined as $T_1/(T_PP)$,



Figure 2: FMM speedup for vortex panels

where T_1 is the time for a single-processor simulation and T_P is the time for the corresponding simulation using Pprocessors. Note that efficiency is better than 80% for 10⁵ particles on 8 processors, and better than 90% for 10⁶ particles. On 4 processors, the efficiency is better than 90% even for 10⁵ particles. The efficiency appears to be poor for 10⁴ particles on 4 and 8 processors, and not too impressive even on 2 processors. This low efficiency at smaller number of particles is somewhat irrelevant, especially when the wall clock time is very small and since the number of computational elements in practical simulations jumps to 10⁵ within a few timesteps. The more important property is the increase in the efficiency to over 90% with the increase in the number of particles.

A limited number of tests have been conducted using external flow over a sphere to verify the robustness of the



Figure 3: FMM parallelization efficiency

proposed algorithm in the laminar flow regime. Flow over a sphere is a particularly challenging canonical problem that can clearly demonstrate the robustness of an algorithm, because it undergoes various flow regime changes within a narrow range of Reynolds numbers (such as steady axisymmetric flow, steady non-axisymmetric flow, periodic flow, and others). To date, we have completed flow simulations for up to $\text{Re}_D = 100$, which is in the steady axisymmetric flow regime, and have begun a simulation at $\text{Re}_D = 250$, which is a steady but asymmetric flow. These flow regimes are difficult to capture accurately with vortex methods, because unless the diffusion solver is robust and can damp out the growth of instabilities, the simulations may predict unphysical flow unsteadiness and/or asymmetry.

For simulations at $\text{Re}_D \leq 100$, a set of 1,280 triangular panels is used to discretize the surface of the sphere, and the computational timestep is set to $\Delta t = 0.025$. Results with $\Delta t = 0.01$ provided similar flow fields. To curb the linear growth of the number of particles, the computational domain is extended to at least four diameters downstream of the sphere; beyond that point the particles are simply removed from the calculation. Furthermore, the cutoff circulation below which vorticity is not diffused is set to $|\vec{\Gamma}|_{cutoff} = 10^{-5}$.

Figure 4 depicts the separation angle (top), recirculation bubble size (middle), and the coordinates of the bubble center (bottom) as a function of the Reynolds number as predicted by our method and by others. The results are in very good agreement with predictions by finite difference [21,22] and spectral element methods [23], as well as experimental data [24,25]. Note that in this flow range, the simulations display no unsteadiness and the flow remains axisymmetric for the duration of the simulation, despite the fact that vortex particles are distributed non-uniformly and that *no particle remeshing* is used during the run. The flow axisymmetry essentially implies that it is topologically a two dimensional problem and that the growth of streamwise vortices due to vorticity stretch is properly damped out by the diffusion process. This is a welcome verification of the robustness of the various components used in the current algorithm. In particular, it is emphasized again that unlike the PSE-based simulation of flow over a sphere [6], which requires remeshing once every few (about five) timesteps to maintain solution accuracy, the current VRM-based simulation uses no remeshing.

Simulation of flow over a sphere at $\text{Re}_{D} = 250$, which is in the steady asymmetric regime, is in progress. For this case, the spherical surface is discretized using 1,280 triangular panels, and the timestep is set to $\Delta t = 0.015$. The flow is perturbed laterally (in the cross-stream direction) using one sinusoidal cycle with non-dimensional period T = 1 and amplitude equal to the freestream velocity. Figures 5a and 5b represent a post-perturbation snapshot in time of the particle positions in the planes of symmetry and perturbation, respectively, color-coded with the strain rate. The simulation has yet to reach steady state and the results are only meant to demonstrate the salient features of the proposed method. Notice first that the wake remains symmetric in the (so-called) symmetry plane, the cross-stream plane that is orthogonal to the plane of perturbation. However, it develops two counterrotating bubbles of unequal size in the perturbation plane; the larger bubble being closer to the sphere, in agreement with

experimental evidence. The bubble size is roughly 30 percent smaller than expected, which is not surprising since the flow has not reached steady state yet. Figure 5 clearly shows how vortex particles cluster themselves adaptively to regions with significant vorticity and strain rate. Also notice that small secondary vortices develop adjacent to the surface of the sphere



Figure 4: Comparison of flow over a sphere

and immediately downstream of where the boundary layers separate from the sphere.

It is worth mentioning here that we have actually tried perturbing the flow for the $\text{Re}_D \le 100$ cases, and have observed that the flow returns back to its stable, symmetric condition. Also, for the $\text{Re}_D = 250$ case the perturbed flow returns to a condition of full symmetry when relatively large timesteps (e.g., $\Delta t = 0.025$) are used in the prediction. This makes sense since, in an under-resolved simulation, using a large timestep is roughly equivalent to assigning a small Reynolds number.



Figure 5: Side views of the strain rate field at (a) plane of symmetry, and (b) plane of perturbation

CONCLUSION

A hybrid Lagrangian Vortex-Boundary Element and Vorticity Redistribution Method (VRM) is developed for the grid-free simulation of 3-D incompressible flow about complex objects. Computations are accelerated using a MPI-parallel adaptive Fast Multipole Method (oct-tree code). In this paper, the computational algorithm is described briefly, followed by a demonstration of the CPU speed up obtained due to the implementation of a MPI-parallel adaptive FMM (oct-tree code). The results indicate that our implementation of FMM is as good as or better than the results by other researchers in the field. Finally, the first in a series of benchmark results is presented using the example of flow over a sphere at low Reynolds numbers. Very good agreement with other simulations as well as experimental data is obtained for runs at $\operatorname{Re}_{D} \leq 100$ range. The predictions showed experimentally observed steady flow symmetry, despite the fact that the computational elements are distributed non-uniformly in the fluid domain. This is the first ever long-time vortex-based simulation in 3-D - with preliminary validation of results that confirms the claim that VRM facilitates truly grid-free computations without having to resort to frequent remeshing. Finally, initial results for the case of $\text{Re}_{D} = 250$ indicate that the proposed method appears to capture the physically observed asymmetry of the flow. The latter is only qualitative in nature

at this stage, since the computations have not reached the stationary state yet.

ACKNOWLEDGMENTS

This project was conducted under NSF SBIR Phase II Grant No. 0060554. The first author wishes to thank Dr. Shankar Subramaniam for valuable discussions on the topic of boundary conditions for vorticity.

REFERENCES

- 1 Gharakhani, A., and Ghoniem, A. F., 1996, "Vortex Simulation of the Three-Dimensional Wake Behind A Cube," in *34th Aerospace Sciences Meeting*, AIAA-96-0170, pp. 1-22, Reno, Nevada.
- 2 Gharakhani, A., and Ghoniem, A. F., 1997, "Three dimensional vortex simulation of time dependent incompressible internal viscous flows," J. of Computational Physics, 134, No. 1, pp. 75-95.
- 3 Gharakhani, A., and Ghoniem, A. F., 1997, "Toward A Grid-Free Simulation of Flow In Engines," in *The 1997* ASME Fluids Engineering Division Summer Meeting, FEDSM97-3025, pp. 1-11, Vancouver, Canada.
- 4 Gharakhani, A., and Ghoniem, A. F., 1998, "3D Vortex Simulation of Flow In Engines During Compression," in *The 1998 ASME Fluids Engineering Division Summer Meeting*, FEDSM98-5000, pp. 1-9, Washington D.C.
- 5 Gharakhani, A., and Ghoniem, A. F., 1999, "3-D Vortex Simulation of Flow In An Opposed-Piston Engine," in Proceedings of the Third International Workshop on Vortex Flows and Related Numerical Methods, Toulouse, France, ESAIM: Proceedings, 7, pp. 161-172, http://www. edpsciences.org/articlesproc/Vol.7/abs/gharakhani.html.
- 6 Ploumhans, P., Winckelmans, G. S., Salmon, J. K., Leonard, A., and Warren, M. S., 2002, "Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at Re = 300, 500, and 1000," *J. of Computational Physics*, **178**, pp. 427-463.
- 7 Degond, P., and Mas-Gallic, S., 1989, "The weighted particle method for convection-diffusion equations. Part I: the case of an isotropic viscosity. Part II: the anisotropic case," *Mathematics of Computation*, **53**, pp. 485-525.
- 8 Barba, L. A., Leonard, A., and Allen, C. B., 2005, "Advances in viscous vortex methods – meshless spatial adaption based on radial basis function interpolation," *International J. for Numerical Methods in Fluids*, 47, pp. 387-421.
- 9 Gharakhani, A., 2001, "Grid-Free Simulation of 3-D Vorticity Diffusion By A High-Order Vorticity Redistribution Method," in 15th AIAA Computational Fluid Dynamics Conference, AIAA-2001-2640, pp. 1-10, Anaheim, CA.
- 10 Shankar, S., and van Dommelen, L., 1996, "A new diffusion procedure for vortex methods," J. of Computational Physics, 127, pp. 88-109.

- 11 Shankar, S., 1996, "A New Mesh-Free Vortex Method," *Ph.D. Thesis*, The Florida State University.
- Cottet, G.- H., 1996, "Artificial viscosity models for vortex particle methods," *J. of Computational Physics*, 127, pp. 299-308.
- 13 Gharakhani, A., 2002, "A grid-free method for LES of incompressible flow," in *The 2002 ASME Fluids Engineering Division Summer Meeting & Exhibition*, FEDSM2002-31370, Montreal, Canada.
- 14 Beale, J. T., and Majda, A., 1981, "Rates of Convergence for Viscous Splitting of the Navier-Stokes Equations," *Mathematics of Computation*, 37, No. 156, pp. 243-259.
- 15 Wang, Q. X., 2004, "Variable Order Revised Binary Treecode," J. of Computational Physics, 200, pp. 192-210.
- 16 Gharakhani, A., and Ghoniem, A. F., 1997, "BEM Solution of the 3D Internal Neumann Problem and a Regularized Formulation for the Potential Velocity Gradients," *International J. for Numerical Methods in Fluids*, 24, No. 1, pp. 81-100.
- 17 Koumoutsakos, P., Leonard, A., and Pepin, F., 1994, "Boundary conditions for viscous vortex methods," J. of Computational Physics, 113, pp. 52-61.
- 18 Strickland, J. H., Gritzo, L. A., Baty, R. S., Homicz, G. F., and Burns, S. P., 1999, "Fast multi-pole solvers for three-dimensional radiation and fluid flow problems," in *Proceedings of the Third International Workshop on Vortex Flows and Related Numerical Methods*, Toulouse, France, August 1998; ESAIM Proceedings, 7, pp. 408-417; http://www.edpsciences.org/articlesproc/Vol.7/abs/ strickland.html.
- 19 Warren, M. S., and Salmon, J. K., 1995, "A portable parallel particle program," *Comp. Phys. Comm.*, 87, pp. 266-290.
- 20 Cheng, H., Greengard, L., and Rokhlin, V., 1999, "A fast adaptive multipole algorithm in three dimensions," J. of Computational Physics, 155, pp. 468-498.
- 21 Johnson, T. A., and Patel, V. C., 1999, "Flow past a sphere up to a Reynolds number of 300," *J. Fluid Mechanics*, **378**, pp. 19-70.
- 22 Magnaudet, J., Riviero, M., and Fabre, J., 1995, "Accelerated flows past a sphere or spherical bubble. Part 1., steady streaming flow," *J. Fluid Mechanics*, **284**, pp. 97-135.
- 23 Tomboulides, A. G., Orszag, S. A., and Karniadakis, G. E., 1993, "Direct and Large Eddy Simulations of axisymmetric wakes," *AIAA paper 93-0546*.
- 24 Pruppacher, H. R., Le Clair, B. P., and Hamielec, A. E., 1970, "Some relations between drag and flow pattern of viscous flow past a sphere and a cylinder at low and intermediate Reynolds numbers," *J. Fluid Mechanics*, 44, part 4., pp. 781-7901.
- 25 Taneda, S., 1956, "Experimental investigation of wake behind a sphere at low Reynolds numbers," *J. Physical Society of Japan*, **11**, No. 10, pp. 1104-1108.