23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

FAST MULTIPOLE DISCRETE VORTEX METHOD APPLIED TO UNSTEADY FLOW SIMULATIONS

Ricciardi, T. R
Wolf, W. R.
Universidade Estadual de Campinas - UNICAMP
Campinas, SP, 13083-600, Brazil
tulioricci@fem.unicamp.br, wolf@fem.unicamp.br

Bimbato, A. M.
Universidade Estadual Paulista - FEG UNESP
Guaratinguetá, SP, 12516,410 - Brazil
alexbimbato@feg.unesp.br

Abstract. *The discrete vortex method (DVM) is based on a Lagrangian description of the vorticity transport equation. However, the solution of the fluid velocity field requires the evaluation of the contributions from the incident flow, the perturbation due to the body and the vortex-vortex interactions. The latter contribution is computationally expensive since the Biot-Savart law is used to compute the induced velocity by all discrete vortices in the cloud. The fast multipole method (FMM) is an attractive algorithm used to accelerate the expensive vortex interactions. It reduces the computational cost of the Biot-Savart law from O(N^2) to O(N), where N is the number of discrete vortices in the cloud for a particular time step. In this work work, the analysis of the flow past an impulsively started cylinder is presented, including the computational time reduction for the FMM.*

Keywords: *Discrete Vortex Method, Fast Multipole Method, Fluid Mechanics, Lagrangian Method*

1. INTRODUCTION

The Discrete Vortex Method (DVM) is a meshless numerical method based on a Lagrangian description of the vorticity transport equation, which is split into diffusive and convective effects (Chorin, 1973). Several formulations can be used to model the diffusive effects (Barba, 2004), *e.g.*, the random walk method, the core spreading method, the particle strength exchange, *etc*. The convection term can be treated using a material derivative to avoid the solution of a non-linear term. Therefore, each vortex is convected with the fluid velocity field, which is evaluated by the contributions from the incident flow, the perturbation due to the body, and the vortex-vortex interactions calculated by the Biot-Savart law.

The DVM is a method used to represent wakes and boundary layers of arbitrary bodies using a discrete vortex cloud. The vortex-vortex interaction requires an expensive convolution step of $O(N^2)$ calculations, which imposes a heavy limitation on the usage of the method to solve engineering problems. With that in mind, alternative ways are required to accelerate the DVM simulations.

The Fast Multipole Method is listed as one of the top 10 algorithms of the twentieth century (Cipra, 2000). It was developed by Greengard and Rokhlin (1987) for the solution of N-body gravitational problems. The algorithm consists of clustering the influence of elements close to each other into multipole expansions, and then evaluating their interaction at distant locations, *i.e.*, the center of far away clusters. This way, the influence among different groups of particles is computed faster than the $O(N^2)$ operations required by the direct Biot-Savart law.

Due to the cost associated with vortex clustering, the reduction in time from $O(N^2)$ to O(N) is achieved when N is sufficiently large and a hierarchical multi-level multipole expansion is used. The method is not applicable among near-field clusters of vortex particles and, hence, these calculations are still performed using the Biot-Savart law.

There are two different multi-level FMM algorithms (Greengard and Carrier (1987); Carrier *et al* (1988)); the first one is suited when the distribution of particles is nearly uniform in the domain. In this case, the computational box is uniformly refined to a prescribed level. The second implementation, which applies an adaptive refinement of the computational boxes, is highly efficient in cases where particles are clustered in specific locations. In the adaptive method, boxes are only refined in regions where clusters of particles exist.

Vortex wakes concentrate behind aerodynamic bodies, so one can argue that the adaptive method provides better cost reductions in this case since empty boxes are neither refined nor used. However, the non-stationary wake developed downstream the body and the continuous vortex shedding mechanism along the body surface by the DVM requires an update of the pre-processing step and the continuous creation of lists of box interactions in the adaptive FMM every time step, considerably increasing the simulation time.

The vortex cloud is, then, concentrated in a small region of the FMM domain, making the analysis expensive due to the Biot-Savart interaction among adjacent boxes. To avoid this cost, it is necessary to use several refinement levels, increasing the memory cost of the method due to the large number of computational boxes in the highest refinement level. This way, efficient ways to allocate memory are necessary.

We use the non-adaptive multi-level FMM with only a few pre-processing steps, along with several techniques to speed up both pre-processing and FMM steps. A further acceleration of the computational simulation is also obtained with parallelization using OpenMP in several steps of the present FMM implementation. The simulation of the flow past an impulsively started cylinder is presented in this work, along with the comparison of the computational time used by both the DVM-FMM and the direct DVM.

2. MATHEMATICAL METHOD

2.1 Discrete Vortex Method

The fluid flow is governed by the continuity equation, Eq. (1), and the unsteady 2D Navier-Stokes equation (N-S equations), which is expressed in terms of the z-vorticity vector, $\boldsymbol{\omega} \equiv (0,0,\omega)$, Eq. (2):

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

$$\frac{\partial \omega}{\partial t} + \vec{u} \cdot \nabla\omega = \frac{1}{Re} \nabla^2\omega. \tag{2}$$

For the numerical simulation, the vorticity in the fluid flow is represented by a cloud of discrete vortices, whose cores are filled with a Gaussian distribution of vorticity. Since the left hand side of the vorticity equation carries on the information about the convection and the right hand side the information about the diffusion, Chorin (1973) proposes the use of the so called viscous splitting algorithm. According to this algorithm, at a time step of the numerical simulation, the convection process is governed by Eq. (4), while the diffusion is governed by Eq. (5):

$$\frac{\partial \omega}{\partial t} + \vec{u} \cdot \nabla\omega = \frac{D\omega}{Dt} = 0, \tag{4}$$

$$\frac{\partial \omega}{\partial t} = \frac{1}{Re} \nabla^2\omega. \tag{5}$$

2.1.1 Convection

In this work, the convection equation is written using a Lagrangian description and its solution is obtained using Euler's time marching method, while the solution of the diffusion equation is obtained using the random walk method. Once the vorticity field is modeled by a cloud of discrete vortices, Eq. (4) is written in Lagrangian form as:

$$\frac{d\vec{x}_j}{dt} = \vec{u}_j(\vec{x}, t), \quad j = 1, NV \tag{6}$$

where NV is the number of vortices in the cloud and the velocity field **u**(**x**,t) can be split in three parts according to Bimbato *et al.* (2009):

$$\vec{u}(\vec{x}, t) = \overrightarrow{ui}(\vec{x}, t) + \overrightarrow{ub}(\vec{x}, t) + \overrightarrow{uv}(\vec{x}, t) \tag{7}$$

The influence of the incident flow is represented by **ui** (**x**,t). For a dimensionless uniform flow with zero angle of incidence, its components are:

$$ui_1 = 1 \text{ and } ui_2 = 0 \tag{8}$$

The velocity **uv**(**x**,t), due to the vortex-vortex interactions using Biot-Savart law, is given by Eq. (9), where γ_j is the *j*-vortex strength and $c_{kj}^i \left[x_k(t) - x_j(t)\right]$ is the *i* component of the velocity induced, at point $\mathbf{x}_k$(t), by a vortex located at $\mathbf{x}_j$ (t) with unit strength.

$$uv_k^i = \sum_{j=1}^{NV} \gamma_j \, c_{kj}^i \left[x_k(t) - x_j(t)\right], \quad i = 1, 2 \tag{9}$$

The body is represented using the Panel Method, with constant sources distributed along straight lines (panels) as described in details by Katz and Plotkin (1991). The indices 1 and 2 denotes the beginning and the end of the panel, respectively. The tangential, Eq. (10), and normal velocities, Eq. (11), are given in the local coordinate system of each panel, where *x* and *z* are the panel longitudinal and normal direction, respectively.

$$u_t = \frac{\sigma}{4\pi} ln\left(\frac{(x - x_1)^2 + z^2}{(x - x_2)^2 + z^2}\right) \tag{10}$$

$$u_n = \frac{\sigma}{2\pi}\left[\tan^{-1}\frac{z}{x - x_2} - \tan^{-1}\frac{z}{x - x_1}\right] \tag{11}$$

The Euler method is applied to the solution of Eq.(6) as:

$$x_k^i(t + \Delta t) = x_k^i(t) + u_k^i(\vec{x}_k, t)\,\Delta t, \quad i = 1,2 \tag{12}$$

2.1.2 Diffusion

The solution to the diffusion process, Eq. (5), in the Lagrangian context, is obtained using the random walk method (Chorin, 1973) to simulate the Brownian motion of the particles. According to this method, the random walk diffusion displacement $\boldsymbol{\eta}(\mathbf{x},t)$, with a zero mean and a $(2\Delta t/Re)$ variance, is added after the convective step. Hence, the position of each vortex at the instant $(t + \Delta t)$ is given by:

$$x_k^i(t + \Delta t) = x_k^i(t) + u_k^i(\vec{x}_k, t)\Delta t + \eta_k^i, \quad i = 1,2 \tag{13}$$

The random displacement components are:

$$\eta_k^1 = \sqrt{\frac{4\Delta t}{Re}\,\log\left(\frac{1}{P}\right)}\cos(2\pi Q), \tag{14}$$

$$\eta_k^2 = \sqrt{\frac{4\Delta t}{Re}\,\log\left(\frac{1}{P}\right)}\sin(2\pi Q). \tag{15}$$

with P and Q being random numbers laying between zero and one.

2.2 Aerodynamic Coefficients

With the vorticity field, it is possible to obtain a Poisson equation for the pressure and its solution is given through the following integral formulation (Shintani and Akamatsu, 1994):

$$H\overline{Y}_p - \int_{S_1}\overline{Y}\nabla\Xi_p\cdot\boldsymbol{e}_n dS = \iint_{\Omega}\nabla\Xi_p\cdot\left(\overline{\boldsymbol{u}}\times\overline{\boldsymbol{\omega}}\right)d\Omega - \frac{1}{Re}\int_{S_1}\left(\nabla\Xi_p\times\overline{\boldsymbol{\omega}}\right)\cdot\boldsymbol{e}_n dS \tag{16}$$

where the pressure is computed at the *p-th* pivotal point, *H = 1* in the fluid domain, *H = 1/2* at the boundaries, Ξ is a fundamental solution of Laplace Equation, $\overline{Y}$ is the specific work and e_n is the unit vector normal to each source flat panel. The drag and lift coefficients are obtained by pressure integration:

$$C_l = \sum_{j=1}^{Np} CP_j\,\Delta s_j\,\hat{n}_{j,y}, \tag{17}$$

$$C_d = \sum_{j=1}^{Np} CP_j\,\Delta s_j\,\hat{n}_{j,x}. \tag{18}$$

2.3 Fast Multipole Method

In order to reduce the $O(N^2)$ computational cost of the DVM, alternative ways are necessary. The FMM was initially introduced by Greengard and Rokhlin (1987), and improved by Carrier *et al* (1988). Its main steps are: the creation of boxes in a pre-processing step, the clustering of particles into multipole expansions, calculated in the finest level of refinement, *i.e.*, the smaller boxes, (step 1); the creation of multipole expansions at the parent (larger) boxes up to level 2 of refinement (step 2); the computation of multipole interactions among distant clusters (step 3); the translation of far-field interactions, calculated in the previous step, to the children boxes (step 4); and, finally, the computation of the near-field interactions at the highest refinement level (step 5).

The main difference between the global and the adaptive FMM is how the boxes are created. In the first case, parent boxes at level *l* are always divided in 4 children boxes at level *l+1*, and the refinement is independent of the particles. In the second case, boxes at level *l* with a large number of particles are refined to level l+1, and its children are the non-empty boxes, allowing that the number of children to be smaller than 4.

In the adaptive FMM, the motion of the vortices requires the boxes to be constantly refined to adapt to local clusters, so the pre-processing have to be updated every time step. On the other hand, the global method only depends of the total number of particles, being independent of the local clusters. So it is possible to make the preprocessing only when a particle leaves the domain or when the total number of particles becomes too large. This way, by avoiding constant pre-processing, it is possible to have a faster simulation compared to the adaptive FMM.

In this work, a global refinement is done, but the empty boxes are neglected (following the ideas from the adaptive method). Also, the size of the initial box in the computational domain and its maximum refinement level are imposed. The domain is increased whenever a vortex leaves it. When the total number of vortices becomes too large, refinement is applied in order to reduce the near-field interaction cost.

2.3.1 Pre-processing 1

The pre-processing 1 creates variables related to the boxes which are completely independent of the particles. This step can be done only when it is necessary to increase the domain or its maximum refinement level. The memory allocation requires both the number of levels and the total number of boxes. So, efficient ways to allocate the variables are implemented to avoid unnecessary memory allocation, *e.g.*, derived data type.

Lists of well-separated boxes and their neighbors are also determined here. An important characteristic of the global refinement is that the boxes are similar to a regular structured mesh, so it is easy to know which are the adjacent boxes. It is possible to create the two interaction lists simultaneously: denoting *B* as the parent of a box *b*, *B* can have up to 9 boxes sharing a node, being 8 neighbors and *B* itself. These 9 boxes have 36 children boxes, which can be classified as: *b* itself (one box), the near neighbors of *b* (8 boxes) or the 27 well-separated boxes in the interaction list of *b*.

2.3.2 Pre-processing 2

This step determines all the particles inside a box and also the box for every particle. The initial box can be divided into 4 identical children boxes in level 1 of refinement, and, by the relative position of the particle to the center of the level 0 box, it is possible to discover which of the 4 boxes contain the particle at level 1. This process can be repeated up to the penultimate level, resulting in the particle's finest level box.

Simultaneously, we obtain the list of all non-empty boxes and the list of the particles inside each of these boxes. Both lists make the memory allocation more efficient and also reduces the analysis time in the loops of steps 1 to 4.

2.3.3 Particle-to-multipole: step 1

The creation of multipole expansion, *a*, in the center of the box *b*, in the finest level *L*, is given by Eq. (19) as a Taylor's series truncated after *p* terms, for *n* particles with intensity γ_i, with a complex distance of z_i from the center of the box: x_i and y_i are the real and imaginary components of z_i, respectively (see Eq. (20)).

The sum of all vortex particle intensities is given by Eq. (21). In this case, the series is a logarithmic function (vortex interaction), and for different applications, one should change the functions in the following equations. The error in FMM is strictly related to the truncation term, and the lower the number of terms in the series, the larger the error.

$$a(b,k,L) \equiv \left(\sum_{i=1}^{n}(-\gamma_i)\left(\frac{{z_i}^k}{k}\right)\right), \qquad k = 1,p \tag{19}$$

$$z_i = (x_i - x_{box}) + j(y_i - y_{box}), \tag{20}$$

$$Q(L) \equiv \sum_{i=1}^{n}\gamma_i. \tag{21}$$

2.3.4 Multipole-to-multipole: step 2

After the finest level, multipole expansions *a(l)* are created and the influences are translated to the center of the parent box *B*, at the level *l-1*, resulting in *a(B, k, l-1)* in Eq. (21). These operations are done for every *k*-term in a Taylor series with *p* terms. The intensity of the multipoles in level *l-1*, *Q(l-1)* in Eq. (23), is the sum of the children intensities, *Q(l)*. In the inner loop, it is necessary to use the binomial coefficient given by *k* and *kk*.

$$a(B,k,l-1) \equiv \sum_{i=1}^{4}\left(\left(\sum_{kk=1}^{k}\left(a_i(kk,l)z_i^{k-kk}\binom{k-1}{kk-1}\right) - Q_i(l)\frac{z_i^k}{k}\right), \quad k=1,p\right) \tag{22}$$

$$Q(l-1) \equiv \sum_{i=1}^{4} Q_i(l) \tag{23}$$

2.3.5 Multipole-to-local: step 3

The multipole-to-multipole (M2M) steps are made up to level 2, which is the lowest level possible to have multipole-to-local calculations (M2L). The interaction list of a box contains all the 27 source boxes *j* that will interact via M2L with objective box *B*, resulting in the M2L variable b(*B*, *kk*, *l*). The M2L step is given by Eq. (24):

$$b(B,kk,l) \equiv \sum_{j=1}^{nbox}\left(\left(\frac{\sum_{k=1}^{p}\left(a_j(k,l)\left({}^{-1}/_{z_j}\right)^k\binom{kk+k-1}{kk-1}\right)}{z_j^{kk}} - \frac{Q_j(l)}{kk\, z_j^{kk}}\right), \quad kk=1,p\right) \tag{24}$$

where *nbox* represents all the non-empty boxes from the interaction list of *B*. In this equation, z_j is the complex distance between the box with the multipole representation to that with the local representation, and *a* and *Q* are obtained from the previously calculated multipole expansion from box *j* in the interaction list of *b(l)*.

2.3.6 Local-to-local: step 4

The local-to-local (L2L) step represents the translation of the influence coefficients from the boxes that are in the interaction list of the *b*'s parent box, *B*, to a child box *b*. This way, the influences of boxes at coarser levels than *b*'s level are evaluated in this step. The influence of all distant boxes of *b* is the sum of the M2L steps from the boxes of its own interaction list and also from the L2L steps from its parent, *B*.

The L2L influence in a box in the level *l+1* from its parent, at level *l*, is given by Eq. (25). Here, *b(k, l)* is the local representation of the far field multipole expansions at the parent box and z_i is the complex distance from the parent's center to its children.

$$c(b,k,l+1) \equiv \sum_{i=1}^{4}\left(\sum_{kk=k}^{p}\left(b(k,l)(-z_i)^{kk-k}\binom{kk}{k}\right), \quad k=1,p\right) \tag{25}$$

2.3.7 Local-to-particle: step 5

When all contributions are evaluated in a box, they are translated by another Taylor's series to all particles within the box. Hence, all the interactions from distant particles are evaluated with the steps M2M, M2L and L2L previously shown, while the near-field interactions are computed through the Biot-Savart law.

Equation (26) gives the translation from the center of a box to a particle, resulting in the induced velocity by the near field and distant particles in the vortex *i*:

$$uv_i = V_{nearfield} + \sum_{k=1}^{p} b(k,l)\, k\, (z_i)^{k-1} \tag{26}$$

In this equation, *p* is the truncation term in the Taylor series, *b(l)* is the sum of L2L and M2L in a box in the highest refinement level *l* and z_i is the complex distance from the particle *i* to the center of the box.

In the near-field interaction, we use a vortex with a Gaussian distribution of vorticity to avoid the singularity from a potential vortex and, for far-field calculations with the FMM, we use a simple potential vortex.

3. RESULTS AND DISCUSSION

An initial study of the viability of the FMM together with the DVM was presented by Ricciardi *et al*. (2015). With that in mind, we studied the flow past an impulsively started cylinder.

3.1 Simulation Time

The process of vortex generation changes the simulation time different every time step. The size of the domain in the FMM also changes whenever a vortex particle leaves it, together with the refinement level (whenever the total number of vortex becomes too large). In the present study, the truncation term is 30 for the Taylor's series. This way, in order to have an optimum simulation time, the exact time to refine the box is initially studied. After several executions, we propose a "rule-of-thumb" which consists in increasing by one the maximum refinement level when the near-field cost becomes 3 times the far-field cost. By increasing the refinement level, it is possible to see in Figure (1) the time reduction in the near-field cost compared to the far-field, but sacrificing this last cost. So, when the near-field becomes expensive compare to the far-field, there is a refinement to reduce total computational time.

All simulations are performed using an Intel Xeon, 2.2GHz, parallelized in 3 thread, with 16Gb of memory.

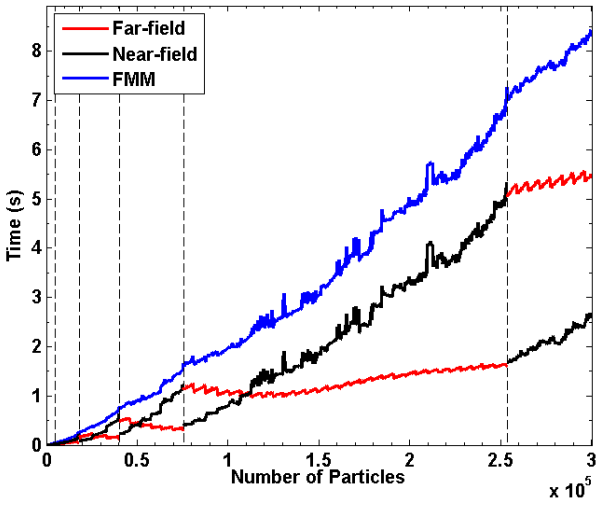

Figure 1 - Wall time for a single time step for different numbers of particles in the wake.

The FMM simulation time is compared to the Biot-Savart time, in Fig. 2a, which is plotted in log-scale. It is possible to see that our implementation is something between O(N) and O(N²). By trial and error, it is about $O(N^{1.3})$. Even for a low number of particles, the FMM is faster than the Biot-Savart method, indicating a very efficient algorithm. The total time of simulation for 1500 time steps is estimated for both methods, and plotted again in log-scale in Fig. 2b. For the FMM, it is extrapolated after 1000 steps, and for the Biot-Savart, it is extrapolated after 300 steps.

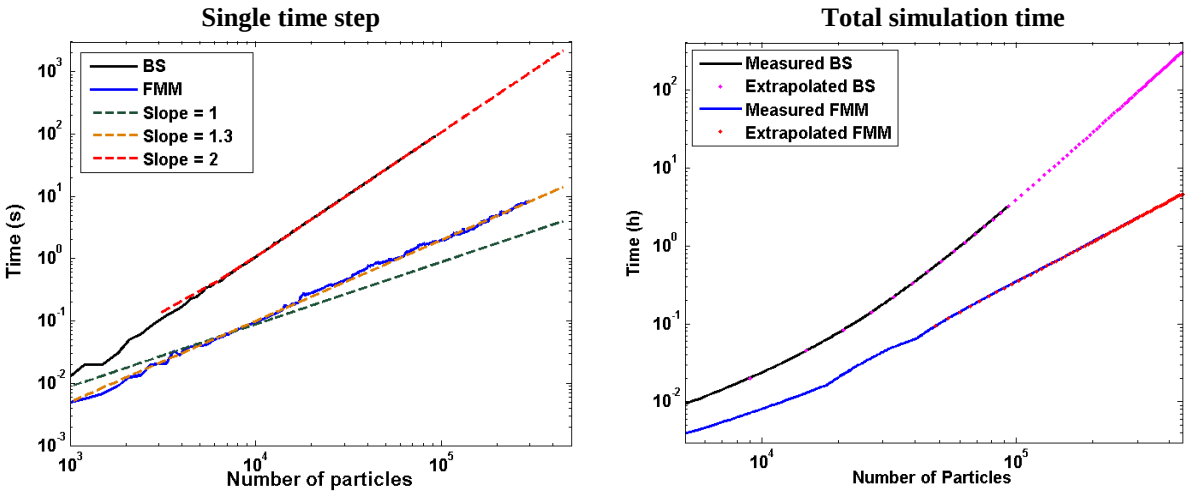

Figure 2 - Comparison of the wall time for FMM and Biot-Savart

3.2 Results

3.2.1 Aerodynamic coefficients

After 2500 time steps, the drag and lift coefficients are calculated using Eqs. (14) to (17) and are shown in Figs. 4a and 4b. The time-marching method used is Euler's method, with a time step equal to 0.05. There are 300 panels to represent the body. The final number of vortex particles is 750.000 and the simulation takes about 14h with the hardware described in section 3.1. The mean values, in red, are calculated in the range between the dashed lines.

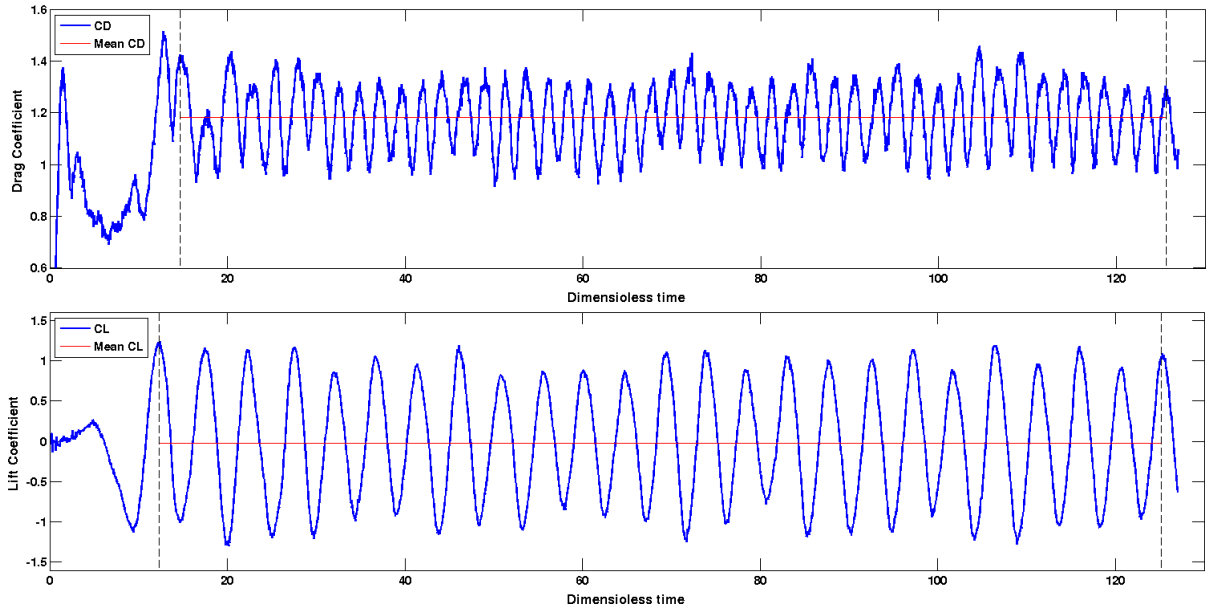

Figure 4 - Aerodynamic coefficients

The Strouhal number indicates the frequency of vortex shedding and it is calculated with a Fast Fourier Transform using Matlab (Figure 5) for the lift coefficient. The results for a single simulation, illustrated in Fig.6 by positive vorticity (red dots) and negative vorticity (blue dots), are compared with experimental data from Lienhard (1966) and are in Table 1, showing a good agreement.

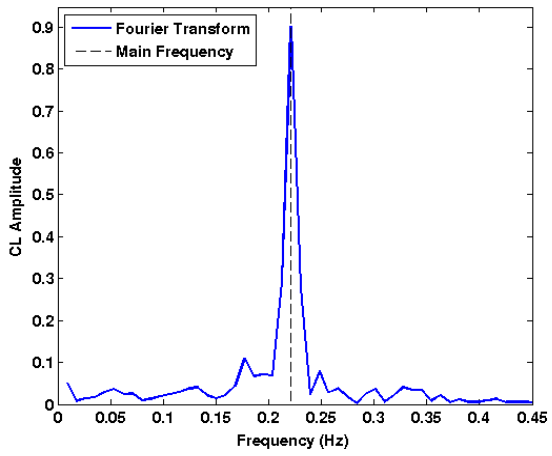

Figure 5. Strouhal Number

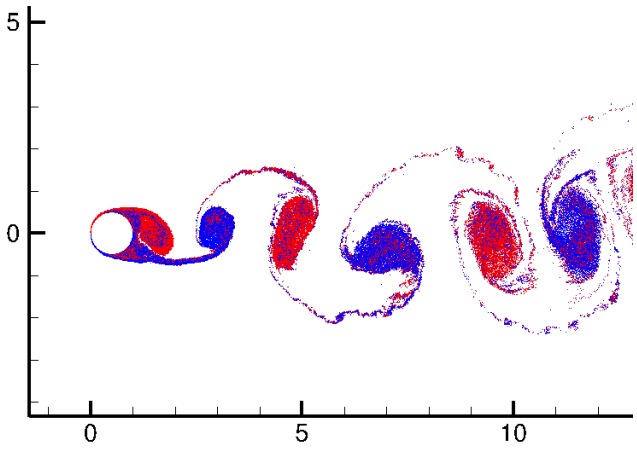

Figure 6. Wake representation by discrete vortex

Table 1. Aerodynamic coefficients

Parameter	FMM	Biot-Savart	Lienhard (1966)
Mean Lift	-0.025	-0.0101	0
Mean Drag	1.1809	1.2157	~ 1.2
Strouhal number	0.2214	0.2226	0.195 ~ 0.215

4. CONCLUSIONS AND FUTURE WORK

In this work we developed a fast numerical tool for the simulation of unsteady high-Reynolds number flows. The Fast Multipole Method showed great efficiency to reduce the total time of analysis compared to the conventional Biot-Savart law from the Discrete Vortex Method alone.

In the future, we will study more accurate time-marching methods, like the Adams-Bashfort class of schemes. Also, together with the vortex-vortex evaluation using the FMM, developments in the algorithm, *e.g.*, the direct solution of the linear system, will allow the solution of multiple bodies.

This tool can be further developed to solve the energy equation and simulate heat exchangers, or use Howe and Powell analogies to solve aeroacoustics problems.

5. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the partial support for this research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grant No. 470695/2013-7. The work was also supported by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under the Research Grant No. 2013/03413-4. The computational resources provided by Centro Nacional de Processamento de Alto Desempenho, CENAPAD-SP, are also gratefully acknowledged. The first author is supported by a CAPES MSc. scholarship, which is also acknowledged.

6. REFERENCES

Barba, L. A., 2004. “Vortex Method for computing high Reynolds number flows: Increased accuracy with a fully mesh-less formulation”, Ph. D. Thesis - California Institute of Technology.

Bimbato, A. M., Alcântara Pereira, L. A., Hirata, M. H., 2009. “Simulation of Viscous Flow around a Circular Cylinder near a Moving Ground”, *J. of the Braz. Soc. of Mech. Sci & Eng.*, v. XXXI, n. 3, July-September, pp. 243-252.

Bimbato, A. M., 2012. “Study of Turbulent Flows around a Smooth or a Rough Bluff Body using the Discrete Vortex Method”, Ph.D. Thesis - Mechanical Engineering Institute, Federal University of Itajubá, 165p. (in Portuguese).

Carrier, J., Greengard, L., and Rokhlin, V., 1988. “A Fast Adaptive Multipole Algorithm for Particle Simulations,” SIAM Journal on Scientific and Statistical Computing, Vol. 9, No. 4, pp. 669–686.

Chorin, A. J., 1973. “Numerical Study of Slightly Viscous Flow”, *Journal of Fluid Mechanics*, v. 57, pp. 785-796.

Cipra, B. A., 2000. “The Best of the 20th Century: Editors Name Top 10 Algorithms,” SIAM News, Vol. 33, pp. 1–2.

Greengard, L., and Rokhlin, V., 1987. “A Fast Algorithm for Particle Simulations,” Journal of Computational Physics, Vol. 73, pp. 325–348.

Katz and Plotkin, 1991. “Low Speed Aerodynamics: from Wing Theory to Panel Methods”, McGraw Hill Inc

Lienhard, J. H., 1966. "Synopsis of Lift, Drag, and Vortex Frequency Data for Rigid Circular Cylinders", Washington State University.

Lewis, R. I., 1991. “Vortex Element Method for Fluid Dynamic Analysis of Engineering Systems”, Cambridge Univ. Press, Cambridge, England, U.K.

Ogami, Y., Akamatsu, T., 1991. “Viscous Flow Simulation Using the Discrete Vortex Model - The Diffusion Velocity Method”, Computers & Fluids, 19, n. 3, pp. 433-441.

Ricci, J. E. R., 2002. “Simulação Numérica do Escoamento ao redor de um Corpo de Forma Arbitrária, Estacionado nas Imediações de uma Superfície Plana, com o Emprego do Método de Vórtices”, Ph.D. Thesis - Mechanical Engineering Institute, Federal University of Itajubá. (in Portuguese)

Ricciardi, T. R., Bimbato, A. M., Wolf, W. R., 2015. “Numerical Simulation of Vortex Interactions Using a Fast Multipole Discrete Particle Method”, 1st Pan-American Congress on Computational Mechanics

Rossi, L. F., 1996. “Resurrecting Core Spreading Vortex Methods: A New Scheme that is both Deterministic and Convergent”, *SIAM J. Sci. Comput.*, vol. 17, pp. 370-397.

Shintani, M., Akamatsu, T., 1994. “Investigation of Two Dimensional Discrete Vortex Method with Viscous Diffusion Model”, Computational Fluid Dynamics Journal, v. 3, n. 2, pp, 237-254

Wolf, W. R., 2011. “Airfoil Aeroacoustics: LES and Acoustic Analogy”, Ph.D. Thesis – Aeronautics and Astronautics, Stanford University

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.