EPTT-2022-0036
EVALUATION OF THE USE OF A FAST MULTIPOLE METHOD IN LAGRANGIAN SIMULATIONS

Marília Fernandes Vidille
São Paulo State University (UNESP), School of Engineering and Sciences, Guaratinguetá, SP, Brazil
marilia.f.vidille@unesp.br

Luiz Antonio Alcântara Pereira
Mechanical Engineering Institute, Federal University of Itajubá (UNIFEI), Itajubá, MG, Brazil
luizantp@unifei.edu.br

Alex Mendonça Bimbato
São Paulo State University (UNESP), School of Engineering and Sciences, Guaratinguetá, SP, Brazil
alex.bimbato@unesp.br

Abstract. *This work shows the association of a fast multipole method with a Lagrangian method to simulate the flow around a circular cylinder. The discrete vortex method is the lagrangian computational technique used and it is characterized by the use of Lamb discrete vortex to represent the vorticity field. To simulate the advection in the discrete vortex method, it is necessary to know the velocity field in the position occupied by each discrete vortex used to discretize the vorticity field. This velocity field suffers three influences: the solid boundary, incident flow and vorticity field. The vorticity field contribution is taken into account by the vortex-vortex interaction, which is especially onerous when the Biot-Savart law is used: the computational costs is proportional to Z^2, where Z is the amount of discrete vortices in the computational domain. So, the use of the Biot-Savart law makes the simulation prohibitive due to the CPU time. Therefore, the aim of this research is to use an accelerator algorithm, the fast multipole method, to reduce the vortex-vortex interaction. The main idea of the method is to divide the computational domain into square boxes, starting from a box of zero refinement level, which contains all discrete vortices presented in the computational domain. Then this initial box is divided into four equal boxes for the first refinement level. These boxes are then divided again into sixteen boxes for the second refinement level and so on. The main idea of the method is that the interaction occurs primarily between the boxes rather than between the particles, thereby reducing the use of the Biot-Savart law and, as a consequence, the final CPU time of the simulations.*

Keywords: *accelerator algorithm, fast multipole method, panel method, lagrangian description*

1. INTRODUCTION

Particle Methods were proposed to solve the differential equations used to model problems encountered in Fluid Mechanics in the presence or absence of mass transport. These methods rely on a Lagrangian approach, making them more attractive than Eulerian methods for numerically solving a category of problems. Lagrangian methods can be more attractive than Eulerian methods in situations where the individual tracking of particles or materials is crucial, where large deformations occur, or where there are complex interfaces and variable geometries. They offer a natural way to handle these challenges, providing greater accuracy and flexibility in many cases.

When using the Discrete Vortex Method, a flow property, in this case, vorticity, is discretized into discrete vortices that are tracked along their trajectories throughout the simulation. The models used to discretize vorticity in two dimensions are based on Rankine and Lamb vortices, where the velocities induced from the potential region of the vortex are smoothed in the vortex core. A vortex is defined as a region with non-zero vorticity surrounded by a region with zero vorticity, and vortex elements are necessary to accurately represent the rotational region of the flow.

The principle of the Discrete Vortex Method for numerically simulating viscous fluid flows is the discretization of the vorticity field through the use of discrete vortices. It's worth highlighting the main characteristics of the method: Navier-Stokes Equations are formulated in terms of the vorticity field, not the velocity field; the Helmholtz theorem (Batchelor, 1967) is used to create a correspondence between fluid material particles and vorticity elements, represented through the discretization of the fluid domain into discrete vortices. Thus, discrete vortices undergo advection processes at the same velocity as fluid particles, ensuring a Lagrangian approach to the method; the vorticity vector $\vec{\omega} = \vec{\nabla} \times \vec{u}$

must be defined to obtain the fluid velocity. By integrating the vorticity field, it's possible to determine the velocity field, $\vec{u}$.

Moreover, it's possible to highlight some aspects that hinder the use of the discrete vortex method: the processing time for calculating the induced velocity in the cloud of discrete vortices using the Biot-Savart law is high since Z^2 operations are required for each Z discrete vortices that make up the cloud; viscous effects in a Lagrangian formulation must be accounted for through statistical or deterministic methods, whereas diffusion is more easily computed through mesh methods; the time increments used must be as small as possible since the effects of Lagrangian evolution in time are better computed the smaller the time increments used, which ends up making the simulations more costly in terms of processing time. This set of characteristics prevents the Discrete Vortex Method from being among the most studied and used CFD (Computational Fluid Dynamics) tools.

To decrease computational time regarding the calculation of the flow velocity field, it's possible to use algorithms capable of reducing the computational efforts inherent to vortex-vortex interaction (the costliest step of said calculation), such as the Fast Multipole Method (Greengard and Rokhlin, 1987; Carrier et al., 1988; Nishimura, 2002; Ricciardi et al., 2017a; Ricciardi et al., 2017b; Vidille et al., 2021).

Through this method, the processor only needs to perform $Z \log Z$ operations or, in the best-case scenario, Z operations for each Z discrete vortices present in the cloud (Koumoutsakos, 1993). It's important to emphasize that, although reducing computational time, efforts are made regarding programming, and the calculation of the velocity field is penalized with a numerical error, which must be monitored so that the use of the Fast Multipole Method does not compromise the accuracy of the results.

Therefore, in this work, the use of the fast multipole method is presented, aiming to evaluate the behavior of this accelerator algorithm and to compare its results with the ones obtained through the Biot-Savart law.

2. GOVERNING EQUATIONS

The problem of the flow around a circular cylinder is governed by the continuity and the Navier-Stokes equations (Eqs. 1-2) and the turbulent and incompressible flow of a Newtonian fluid with constant properties over a circular cylinder causes the viscous wake shown in Figure 1 (where Ω is the fluid domain).

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

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

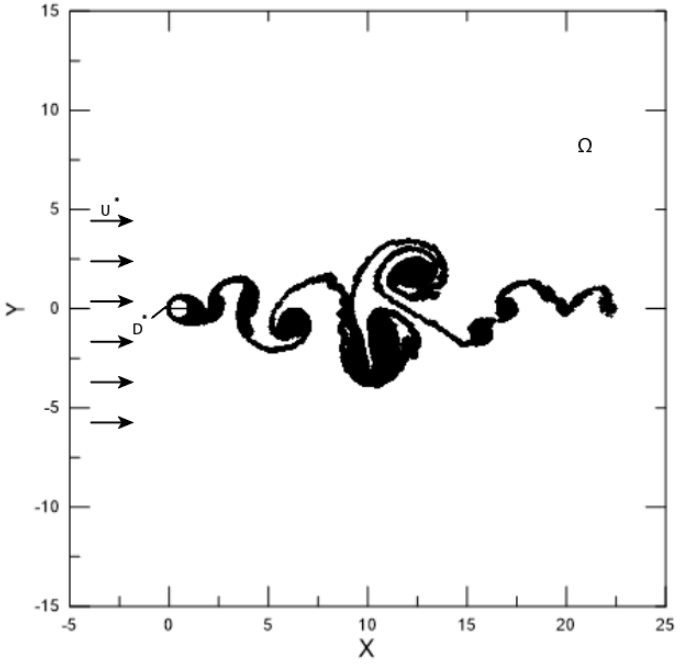


Figure 1. Viscous wake developed downstream of the cylinder and used to evaluate the behavior of the fast multipole method (Re = 1.0 x 10^5).

In Eqs. (1) and (2), the filtered velocity and pressure fields are denoted by an overbar. The Reynolds number is modified ($\mathrm{Re_c}$) to include the effects of turbulence in the flow through a large-eddy simulation and is defined as:

$$\mathrm{Re_c} = \frac{\mathrm{U}^* D^*}{\upsilon + \nu_t} \tag{3}$$

In Eq. (3), U^* is the incident flow velocity, D^* is the circular cylinder diameter, ν is the kinematic viscosity and ν_t is the eddy-viscosity.

By applying the curl operator to the Navier-Stokes equations and considering the continuity equation, ones obtain the vorticity transport equation, which in two-dimensional and simplified form is given by (Batchelor, 1967):

$$\frac{\partial \omega}{\partial \mathrm{t}} + \left(\vec{\mathrm{u}} \cdot \vec{\nabla}\right)\omega = \frac{1}{\mathrm{Re_c}} \nabla^2 \omega \tag{4}$$

From Eq. (4), for convenience in typing, the overbar notation on the variables is omitted.

The Eq. (4) governs the evolution of vorticity, where the left-hand side is responsible for vorticity advection due to temporal variation, while the right-hand side is responsible for vorticity diffusion arising from viscosity effects.

Building upon this information, Chorin (1973) proposed an algorithm to separate the viscous part from the vorticity transport equation. With the use of this algorithm, it is possible to calculate advection and diffusion of vorticity separately within the same time increment. The advection and diffusion of vorticity are given, respectively, by (Chorin, 1973):

$$\frac{\partial \omega}{\partial \mathrm{t}} + \left(\vec{\mathrm{u}} \cdot \vec{\nabla}\right)\omega = \frac{\mathrm{D}\omega}{\mathrm{Dt}} = 0 \tag{5}$$

$$\frac{\partial \omega}{\partial \mathrm{t}} = \frac{1}{\mathrm{Re_c}} \nabla^2 \omega \tag{6}$$

Diffusion is resolved using the random walk method Chorin (1973), which is a probabilistic and easily implementable approach that involves choosing random numbers to compute the radial and circumferential displacements of discrete vortices.

Advection, on the other hand, is resolved by finding the velocity field at the position occupied by each of the discrete vortices within the cloud. This velocity field undergoes three types of influence: incident flow, the solid boundary, and vorticity field.

Regarding the incident flow, this solution is straightforward, as it induces the same velocity in all discrete vortices within the cloud. Concerning the body, the panel method (Katz and Plotkin, 1991) is used, which discretizes the body into panels that can be either flat or curved. Singularities, which can be sources, vortices, or dipoles, are distributed on these panels in a concentrated, constant, or linear manner. For the case of the circular cylinder studied in this work, flat panels are used, beneath which vortical singularities are discretized linearly.

The vorticity field contribution, the final component regarding the calculation of the flow velocity field, is typically performed using the Biot-Savart law, where each discrete vortex within the cloud induces velocity on all others present. The Biot-Savart law is given by:

$$\vec{\mathrm{u}} = -\frac{1}{2\pi} \int_{\Omega} \frac{\vec{\omega} \times \vec{\mathrm{r}}}{|\vec{\mathrm{r}}|^2} \mathrm{d}\Omega \tag{7}$$

where $\vec{u} = u\hat{\imath} + v\hat{\jmath}$ represents the induced velocity at each discrete vortex used to represent the vorticity field, $\vec{\omega}$ denotes the vorticity field, $\vec{r}$ stands for the distance between the Lamb's discrete vortices, and Ω represents the fluid domain.

The use of the Biot-Savart law requires Z^2 processor operations for every Z discrete vortices present in the domain. Therefore, the fast multipole method (FMM) is implemented aiming to divide the computational domain into square boxes and allowing interaction primarily between these boxes rather than between each of the discrete vortices. This enables the CPU time for this step to be proportional to $ZlogZ$ or, ideally, to Z (Koumoutsakos, 1993). The FMM is listed as one of the top 10 algorithms of the twentieth century (Cipra, 2000).

As mentioned earlier, in the fast multipole method, it is necessary to divide the computational domain into square boxes, starting from the first refinement level, which contains four square boxes, up to the last refinement level, which should be determined based on the complexity and quantity of discrete vortices in the simulation. This division of the computational domain into square boxes is illustrated in Figure 2.

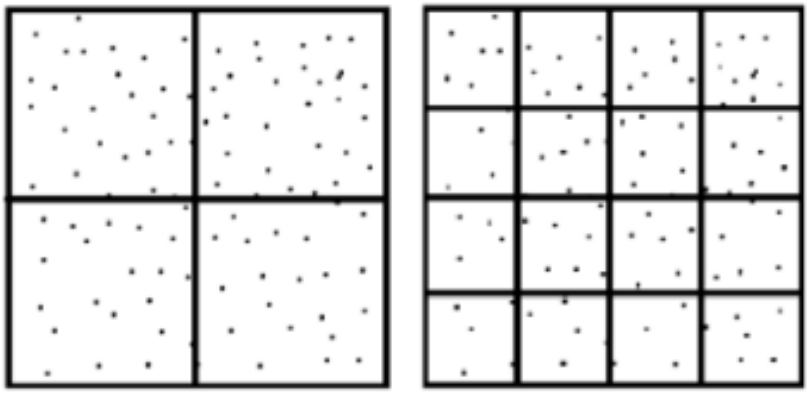


Figure 2. Computational domain divided into square boxes by the fast multipole method.

It can be noted that a refinement level *n* has 4^n boxes. Then, after divide the domain in square boxes, lists of neighbors boxes and well-separated boxes are created. Considering two boxes, *B* and *b*, whereby *b* is a child box of *B*, *B* can have up to 9 boxes sharing a node, being 8 neighbors boxes and *B* itself. Each one of these 9 boxes have 4 child boxes, totaling 36 children box, which can be classified as *b* itself, the near neighbors of *b* (8 boxes) and the 27 well-separated boxes in the interaction list of *b*. After classifying the boxes, the particles are mapped to determine in which box they are at the maximum refinement level.

Then, when the pre-processing step is finished, there are some steps of the fast multipole method that need to be performed. The first step is called particle-to-multipole and consists of creating multipole expansion *a*, in the center of the box *b*, in the finest level *L*, using a Taylor's series truncated after *p* terms, for *Z* discrete vortex with intensity Γ_i, with a complex distance of z_i from the center of the box, represented by Eq. (8). The sum of all vortex particle intensities is given by Eq. (9).

$$a(b,k,L) = \sum_{i=1}^{Z} -\Gamma_i \frac{{z_i}^k}{k} \tag{8}$$

$$Q(L) = \sum_{i=1}^{Z} \Gamma_i \tag{9}$$

The second step is called multipole-to-multipole and consists of passing the influences from the center of the children box, *b*, to the center of the parent box, *B*, at level *l-1*, resulting in $a(B,k,l-1)$, according to Eq. (10). The intensity of the multipoles in level *l-1*, $Q(l-1)$, is the sum of the children's intensities, $Q(l)$, according to Eq. (11).

$$a(B,k,l-1) = \sum_{i=1}^{4} \left\{ \left[\sum_{kk=1}^{k} a_i(kk,l) {z_i}^{k-kk} \binom{k-1}{kk-1} \right] - \left[Q_i(l) \frac{{z_i}^k}{k} \right] \right\} \tag{10}$$

$$Q(l-1) = \sum_{i=1}^{4} Q_i(l) \tag{11}$$

The step whereby the interactions among boxes occur is the multipole-to-multipole which is made up to level 2 because it is the first level that has boxes far enough apart. The interaction list of a box *b* contains, at most, 27 boxes *j* that interact via multipole-to-local with objective box *B*, resulting in the multipole-to-local variable $b(B,kk,l)$. The variable called *nbox* is used to represent the non-empty boxes that are included in the interaction list of *B*, z_j is the complex distance between the box of interest and the box from the interaction list and the variables *a* and *Q* are calculated in the particle-to-multipole and multipole-to-multipole steps.

$$b(B,kk,l) = \sum_{j=1}^{nbox} \left[\frac{a_j(k.l) \left(\frac{-1}{z_j}\right)^k \binom{kk+k-1}{kk-1}}{{z_j}^{kk}} - \frac{Q_j(l)}{kk{z_j}^{kk}} \right] \tag{12}$$

The next step is very similar to the multipole-to-multipole, but here, the influence is passed from *B* to a child box, *b*. This step is called local-to-local and Eq. (13) calculates the local-to-local influence in a box in the level *l+1* from its parent at level *l*; *b(k,l)* is the local representation of the far field multipole expansions at the parent box and z_j is the complex distance between the parent's and the children's centers.

$$c(b,k,l+1) = \sum_{i=1}^{4}\left\{\sum_{kk=k}^{p}\left[b(k,1)(-z_i)^{kk-k}\binom{kk}{k}\right]\right\} \tag{13}$$

The last step can be divided into two: local-to-particle and particle-to-particle. The local-to-particle consists of doing the reverse process of the particle-to-multipole; in other words, one transfers the influence from the center of a box to all particles within the box through another Taylor's series, showed in Eq. (14), whereby ${vv_i}^w$ is the vortex-vortex interaction, *b(l)* is the sum of multipole-to-multipole and local-to-local steps in a box from the highest refinement level (*l*) and z_i is the complex distance between the discrete vortex *i* and the center of its box.

$${uv_i}^w = {vv_i}^w + \sum_{k=1}^{p} b(k,1)k{z_i}^{k-1}, w = 1,2 \text{ and } i = 1,Z \tag{14}$$

The particle-to-particle consists on the interaction between particles of the same box and on the interaction between particles from boxes in the neighbor list through the Biot-Savart law, showed in Eq. (7).

3. RESULTS

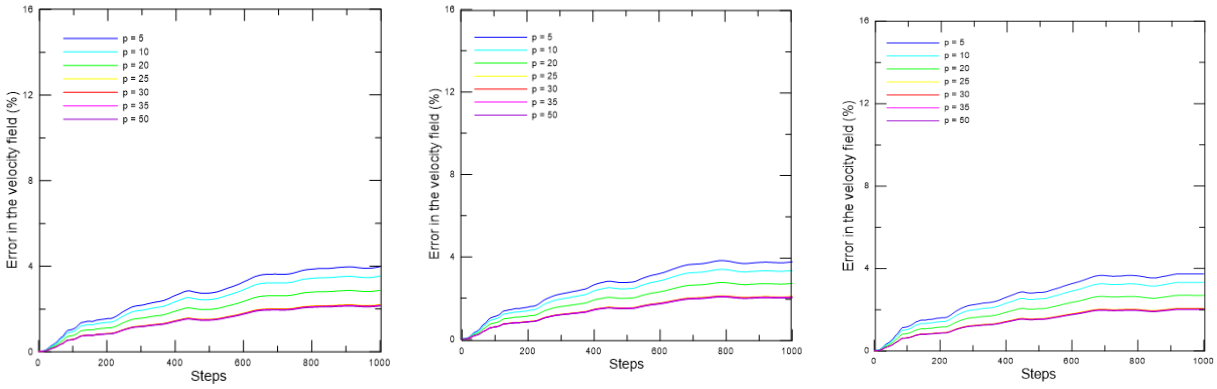
The turbulent and incompressible flow of a Newtonian fluid with constant properties over a circular cylinder causes the viscous wake shown in Figure 1. Such simulation is run taking into account the mathematical formulation presented in Section 2 and is used for the tests of the fast multipole code. As described in Section 2, the fast multipole algorithm divides the computational domain into square boxes (see Figure 2) to reduce the use of the Biot-Savart law (Eq. (7)).

Since the goal is to analyze the efficiency of the fast multipole method, comparing its results with the ones obtained with the Biot-Savart law, the tests are conducted to compute just the contribution of vorticity on the velocity field.

To evaluate all the results obtained with simulations using the acceleration algorithm, one must analyze the numerical error imposed by the Taylor series expansion process in the calculation of the flow velocity field, intrinsic to the fast multipole method. For this purpose, Eq. (15) (Carrier et al., 1988) is employed:

$$\mathrm{E} = \left[\frac{\sum_{\mathrm{i=1}}^{\mathrm{Z}}\left(\mathrm{V_{FMM_i}} - \mathrm{V_{BS_i}}\right)^2}{\sum_{\mathrm{i=1}}^{\mathrm{Z}}\left(\mathrm{V_{BS_i}}\right)^2}\right]^{1/2} \tag{15}$$

Initially, the numerical error associated with the number of terms in the Taylor series (p) for a time increment, Δt, of 0.01 was evaluated, with different maximum refinement levels, namely n = 6, n = 7, and n = 8. Such an error is shown in Figure 3.

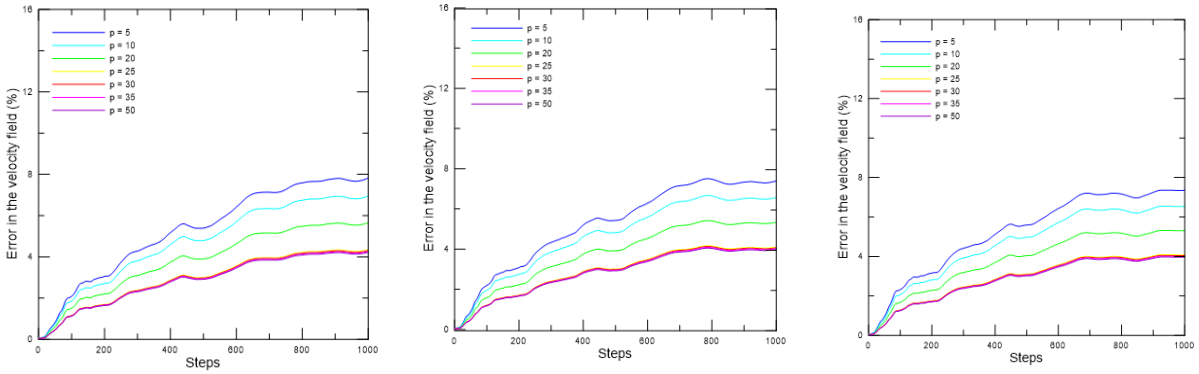


(a) (b) (c)

Figure 3. Numerical error associated with the number of terms used in the Taylor series: (a) n = 6; (b) n = 7; (c) n = 8 (Δt = 0.01).

It is noticeable, for the three maximum refinement levels analyzed (n = 6; n = 7; n = 8), that the numerical error decreases when using a greater number of terms in the Taylor series. Additionally, the errors at the end of the numerical simulations are higher at the maximum refinement level of 6 and decrease for maximum refinement levels of 7 and 8. which is an unexpected result because, as a rule, lower levels of refinement typically reduce numerical error at the end of the simulations, since fewer operations are performed between boxes and more operations are performed directly between particles using the Biot-Savart law. The computational time has been reduced: for the simulation using only the Biot-Savart law, the computational time was 53 hours. In contrast, for 50 terms in the Taylor series, it was 3.9 hours for a maximum refinement level of 6; 2.6 hours for a maximum refinement level of 7, and 2.1 hours for a maximum refinement level of 8.

Then, the numerical error associated with the number of terms in the Taylor series (p) for a time increment, Δt, of 0.025 was also evaluated, with different maximum refinement levels, n = 6, n = 7, and n = 8. Such an error is shown in Figure 4.

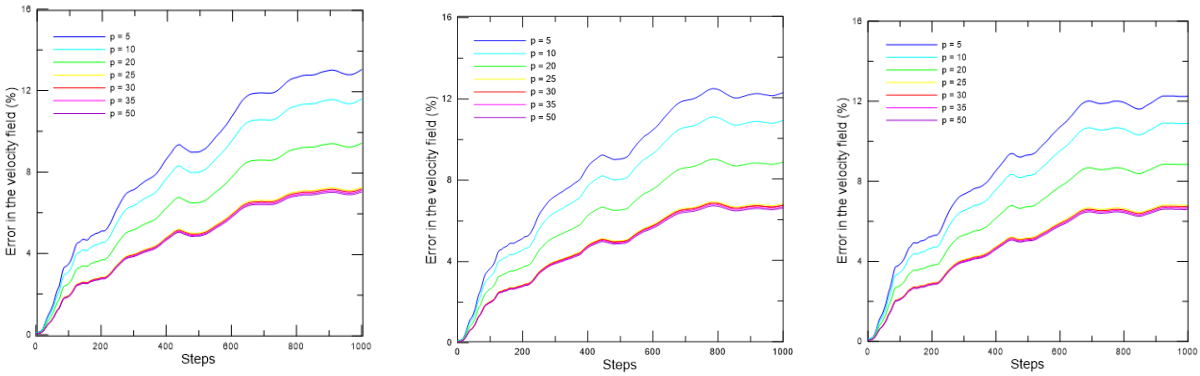


(a) (b) (c)

Figure 4. Numerical error associated with the number of terms used in the Taylor series: (a) n = 6; (b) n = 7; (c) n = 8 (Δt = 0.025).

Again, for a time increment, Δt, of 0.025, also for the three maximum refinement levels analyzed, the numerical error decreases when using a greater number of terms in the Taylor series. Additionally, the errors at the end of the numerical simulations are higher at the maximum refinement level of 6 and decrease for maximum refinement levels of 7 and 8. The computational time has been also reduced: for 50 terms in the Taylor series, it was 3.6 hours for a maximum refinement level of 6; 2.4 hours for a maximum refinement level of 7, and 2.0 hours for a maximum refinement level of 8.

Finally, the last time increment, Δt, analyzed was 0.05. The numerical error associated with the number of terms in the Taylor series (p) for this time increment is shown in Figure 5.



(a) (b) (c)

Figure 5. Numerical error associated with the number of terms used in the Taylor series: (a) n = 6; (b) n = 7; (c) n = 8 (Δt = 0.05).

As expected, also for a time increment, Δt, of 0.05, the numerical error decreases when using a greater number of terms in the Taylor series, p. Additionally, the errors at the end of the numerical simulations are higher at the maximum refinement level of 6 and decrease for maximum refinement levels of 7 and 8.

To better analyze the results, Tab. 1 is observed, which presents the error in the velocity field at the end of the simulations, the CPU time of the simulations, and the reduction in processing time due to the use of FMM, for the number of terms in the Taylor series, *p*, equal to 50, which yielded the best results in all cases.

Table 1. Numerical results obtained with the use of the fast multipole method for all the increment time analyzed (p = 50).

	Refinement level	Error in the velocity field	CPU time (h)	CPU time reduction (%)
Biot-Savart law	-	-	53.00	-
FMM ($\Delta t = 0.01$)	n = 6	2.14	3.93	92.6
	n = 7	2.03	2.61	95.08
	n = 8	2.02	2.14	96.00
FMM ($\Delta t = 0.025$)	n = 6	4.22	3.61	93.19
	n = 7	3.98	2.36	95.55
	n = 8	3.97	2.04	96.15
FMM ($\Delta t = 0.05$)	n = 6	7.03	3.26	93.85
	n = 7	6.63	2.10	96.04
	n = 8	6.61	1.83	96.55

According to Table 1, the computational time for all simulations in which the fast multipole method is used to calculate the velocity field due to vortex-vortex interactions is reduced, with reductions in computational time exceeding 90%. Simulations with a maximum refinement level of 6 have a higher computational time for the three analyzed time increments compared to those with a maximum refinement level of 7, which in turn have higher computational times than those with a maximum refinement level of 8. That is, higher maximum refinement levels allow for lower computational times, as expected (Vidille et al., 2021).

However, for the three analyzed time increments, lower maximum refinement levels result in greater numerical errors in the calculation of the velocity field because there were more interactions between the boxes, via FMM, than directly between the particles, according to the Biot-Savart law. More interactions between boxes than directly between particles cause the numerical error in the velocity field calculation to be higher.

It was expected, according to Vidille et al. (2022), that with higher levels of refinement, numerical errors in the velocity field calculation would be greater. This discrepancy likely occurred due to the positioning of discrete vortices during the numerical simulations. Thus, at lower refinement levels, there probably were fewer empty boxes, leading to more interactions between boxes and increasing the numerical error in the velocity field calculation.

4. CONCLUSIONS

In this work, the focus was on examining how the time increment in numerical simulations and the number of terms in the Taylor series affect the Fast Multipole Method (FMM) algorithm. The simulations conducted, while not involving an excessive number of discrete vortices, were adequate to assess the FMM's efficiency in reducing computational time and maintaining accuracy. The results confirm that increasing the number of Taylor series terms improves the accuracy of the FMM by lowering the numerical error in velocity field calculations.

Tab. 1 illustrates that the FMM substantially cuts CPU time compared to the Biot-Savart law, with reductions more pronounced at higher refinement levels and longer time increments. For example, at the highest refinement level (n = 8) and a time increment of $\Delta t = 0.05$, the CPU time is reduced by 96.55%, while still achieving a low numerical error. This highlights the FMM's effectiveness in balancing computational cost and accuracy.

The findings indicate that more Taylor series terms lead to lower numerical error with the accelerator algorithm. Additionally, contrary to expectations, higher refinement levels also contribute to reduced numerical error, as detailed in the results.

To further investigate the FMM's behavior, future studies could involve more time steps with the same increments. Given that the FMM may accumulate errors, simulations with additional time steps might show higher numerical error by the end of the simulation compared to those conducted up to the thousandth time step.

Future research should expand this analysis to more complex scenarios, such as flow around circular cylinders and aerodynamic profiles, as well as configurations with multiple interacting bodies. These studies will also address situations with more discrete vortices, further evaluating the FMM's robustness and accuracy. The application of FMM to these complex physical scenarios is crucial for the effective use of the discrete vortex method.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge São Paulo Research Foundation (FAPESP), grant 2022/03630-4, National Council for Scientific and Technological Development (CNPq), grant 404539/2023-8, Research Supporting Foundation of Minas Gerais State (FAPEMIG), grant APQ-01246-23 and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), grant 88887.690941/2022-00.

6. REFERENCES

Batchelor, G. K., 1967. An introduction to fluid dynamics. Nova Delhi: Cambridge University Press.

Carrier, J., Greengard, L. and Rokhlin, V., 1988. "Multipole Algorithm For Particle Simulations". *SIAM Journal of Scientific Statistics and Computation*, Vol. 9, No. 4, pp. 669-686.

Chorin, A.J., 1973. "Numerical Study of Slightly Viscous Flow". *Journal of Fluid Mechanics*, Vol. 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, J. and Plotkin, A., 1991. *Low speed aerodynamics: from wing theory to panel methods*. McGraw Hill Inc..

Koumoutsakos, P. D., 1993. *Direct numerical simulations of unsteady separated flows using vortex methods*. Thesis (Ph.D.) - California Institute of Technology.

Nishimura, N., 2002. "Fast multipole accelerated boundary integral equation methods". *Applied Mechanics Reviews*, Vol. 55, pp. 299–324.

Ricciardi, T.R., Wolf, W.R. and Bimbato, A.M., 2017a. "A fast algorithm for simulation of periodic flows using discrete vortex particles". *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 39, pp. 45-55.

Ricciardi, T.R., Wolf, W.R. and Bimbato, A.M., 2017b. "Fast multipole method applied to Lagrangian simulations of vortical flows". *Communications in Nonlinear Science and Numerical Simulation*, Vol. 51, pp. 180-197.

Vidille, M.F., Alcântara Pereira, L.A. and Bimbato, A.M., 2021. "Implementation of an accelerator algorithm to reduce the computational costs in Lagrangian simulations" *In Proceedings of the 26nd International Congress of Mechanical Engineering - COBEM 2021*. Florianópolis, Brazil.

Vidille, M.F., Moraes, P. G., Alcântara Pereira, L.A. and Bimbato, A.M., 2022. "Use of a fast multipole method to reduce the computational costs of aircraft wakes simulations" *In Proceedings of the 19th Brazilian Congress of Thermal Sciences and Engineering – ENCIT 2022*. Bento Gonçalves, Brazil.

Vidille, M.F., Moraes, P. G., Alcântara Pereira, L.A. and Bimbato, A.M., 2023. "The study of the development of aircraft wakes near a ground plane through a lagrangian approach" *In Proceedings of the 27nd International Congress of Mechanical Engineering - COBEM 2023*. Florianópolis, Brazil.

7. RESPONSIBILITY NOTICE

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