19th Brazilian Congress of Thermal Sciences and Engineering
November 06th-10th, 2022, Bento Gonçalves - RS - Brazil

ENC-2022-0027
USE OF A FAST MULTIPOLE METHOD TO REDUCE THE COMPUTATIONAL COSTS OF AIRCRAFT WAKES SIMULATIONS

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

Paulo Guimarães de Moraes
Mechanical Engineering Institute, Federal University of Itajubá (UNIFEI), Itajubá, MG, Brazil
pauloguimor@unifei.edu.br

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

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

Abstract. *The discrete vortex method is a purely lagrangian computational technique, whose objective is to solve the vorticity transport equation, obtained from the Navier-Stokes and the continuity equations. The technique consists of represent the vorticity field using Lamb discrete vortex. To make the solution feasible, the viscous splitting algorithm is used and, according to it, is possible to simulate the advection and diffusion processes separately but in the same time increment. To simulate the diffusion, it is possible to use the traditional random walk method or the core spreading method, whereby the discrete vortices grow until a limit and split in four new particles. To simulate the advection, it is necessary to know the velocity field in the position occupied by each particle and it suffers three influences: incident flow, solid boundary and the discrete vortex cloud. The vortex-vortex interaction is especially onerous, because its computational cost is proportional to* Z^2*, where Z is the amount of particles in the computational domain, when the Biot-Savart law is used. The solid boundary is represented by a source panel method and, in order to satisfy the no-slip boundary condition, at each time of the numerical simulation, a discrete vortex is positioned in the neighborhood of each panel to simulate the boundary layer formation. The increase in the number of particles, because of the core spreading method or the vorticity generation, makes the simulation prohibitive due to the CPU time. Therefore, the aim of the present research is to explore the implementation of an accelerator algorithm, the fast multipole method, who divides the computational domain in square boxes and enables more interactions among boxes than interactions among particles. The physical situation studied in this paper refers to the behavior of vortices detached from airplane wings near the ground. The phenomenon is especially important during the landing and take-off operations, since it may cause accidents in airports. The strategy used consists in analyse the CPU time and the errors in the velocity field computation generated by the fast multipole method when it is compared to the Biot-Savart law calculations. The results show that is possible to reduce the CPU time with acceptable errors which will be important in order to use the discrete vortex method to studied more complex engineering problems.*

Keywords: *lagrangian description, discrete vortex method, fast multipole method, CPU time reduction, panel method*

1. INTRODUCTION

The aeronautical industry is one of the responsible to make the planet globalized. Without it, the international relations would be complex or even be inaccessible. Due to the high demanding on aeronautic industry, it is necessary the development of researches on many subjects, including Fluid Mechanics. Dominate this subject is essential about the safe and competitive development of the aeronautic industry, making the use of resources feasible and optimized.

Due to the growth in the aeronautical industry, the air traffic became high in recent decades, being often the accumulation of landing and take-off operations in a single airport almost simultaneously.

To realize the landing and take-off procedures in the lower time as possible and keeping it safe, it is necessary to analyse the behavior of vortex structures detached from airplane wings, ensuring there is enough time to dissipate the vortices before next airplane use the track. The generation of vorticity from the wings of an airplane is a disturbance that changes the flow velocity field abruptly. It can cause serious accidents to a second aircraft that will operate on the same track several minutes after the first pass. The first requirement for air traffic control at airports is preventing an aircraft from operating in the wake of another (Machol, 1993).

In order to study such important physical situation, the objective of the present research is to perform simulations about the behavior of vortices detached from airplane wings near the ground.

It is possible to solve the problem proposed through two ways: using an eulerian description or a lagrangian description. In the first, the properties of flow field are described as function of their space and time position and the fluid region is discretized through computational meshes, whereby the properties of interest are calculated. In the second technique, the flow properties are treaty how a set of fluid particles followed individually in the fluid domain and the properties of interest are calculated only as a function of the time.

In the present work the authors developed a computational code based on a lagrangian description; the numerical method used is called discrete vortex method (Chorin, 1973; Lewis, 1999; Alcântara Pereira *et al*., 2004; Bimbato *et al*., 2018). The vortex method discretizes a flow property (in this case, the vorticity field) using discrete particles and offers a number of advantages over the traditional eulerian schemes, such as: (i) it is not necessary the use of a mesh to discretize the fluid domain, avoiding instability problems; (ii) it is not necessary to explicitly treat convective derivates; (iii) only the rotational flow regions are solved; (iv) the boundary condition at the downstream end of the flow domain is automatically satisfied.

In the discrete vortex method, the numerical simulation of the vorticity field evolution in time is realized following Lamb discrete particles. Using the viscous splitting algorithm, proposed by Chorin (1973), the solution is divided in two steps: advection and diffusion, which are solved separately but in the same time increment of the numerical simulation. This procedure makes the numerical implementation of the discrete vortex method simplest.

To solve the advection step, it is necessary to know the velocity field of the flow, at each time increment of the numerical simulation, in the position occupied by each discrete vortex used to represent the vorticity field. In the present work, the calculation of velocity field suffers the influences of the solid boundary (airport track) and the discrete vortices cloud (vorticity field). In the last, each discrete vortex induces velocity in all the other ones. This is the more expensive computation in the numerical simulations performed in this work, because it requires Z^2 operations to each Z discrete vortices present in computational domain, when the Biot-Savart law (BS) is used.

Several methods have been proposed to solve the diffusion step and here two are highlighted: the random walk method, proposed by Chorin (1973) and modified by Lewis (1991), whereby the diffusion is simulated through the generation of random values used to compute radial and circumferential displacements of the discrete vortices; the core spreading method, proposed by Leonard (1980) and modified by Rossi (1996), in which the diffusion is simulated by the growth and subsequent partition of the discrete vortices in four new particles, according to previously prescribed parameters that govern the partition process. The first method is probabilistic and it has already been implemented in the computaional code developed by the authors (Vidille *et al*., 2021), but its accuracy is low in comparation with the other diffusion methods, especially when it is compared to the core spreading method. So, with the aim of obtain results closer to the experimental ones available in the literature, the diffusion step of the numerical solution is solved here using the core spreading method.

The use of the diffusion model developed by Rossi (1996) turns the CPU time extremely onerous to determine the flow velocity field, especially when the Biot-Savart law is used to compute the vortex-vortex interaction. As consequence, the use of an accelerator algorithm to calculate the influence of the vortex cloud in the velocity field is mandatory to enable the numerical simulations. For this reason, another contribution of this research is the implementation of the fast multipole method (Greengard and Rokhlin, 1987; Nishimura, 2002; Ricciardi, 2016; Ricciardi *et al*., 2017a; Ricciardi *et al*., 2017b) to reduce the use of the Biot-Savart law, which is the responsible for making the advection step the more expensive stage of lagrangian simulations performed in this work.

The fast multipole method, FMM, is based in the division of the computational domain in square boxes; thus, a cluster of particles is formed from particles that are placed into a box. Therefore, instead of compute all the vortex-vortex interactions, the aim is to compute, mostly, interactions among clusters of particles that are far from each other. The Biot-Savart law is used to calculate the vortex-vortex interactions just among particles that are inside the same box or among particles that are placed in nearby boxes. The FMM is listed as one of the top 10 algorithms of the twentieth century (Cipra, 2000) and the success of its implementation makes the use of the core spreading method feasible to solve the vorticity diffusion equation more accurately.

Finally, it is important to mention that the simulations performed in this work takes into account just the macro scale phenomena, since no turbulence modelling is used.

2. GOVERNING EQUATIONS

In Figure 1 is shown the counterrotating pair of vortices detached from airplane wings, which is represented by Z Lamb discrete vortices with a gaussian distribution of vorticity ; h is the height of the counterrotating pair detachment, b^* is the airplane wingspan, Ω is the fluid domain and S_1 is the solid boundary (in this case, the airport runway, which has a length equal to x_{track}). The circulation of vortex clouds, $+\Gamma^*$ and $-\Gamma^*$ are related to the lift force, F_L, according to the Kutta-Joukowsky law, as in Eq. (1).

$$\Gamma = \frac{F_L}{\rho u^* b^*} \tag{1}$$

In Eq. (1), ρ is the fluid density, u^* is the fluid velocity.

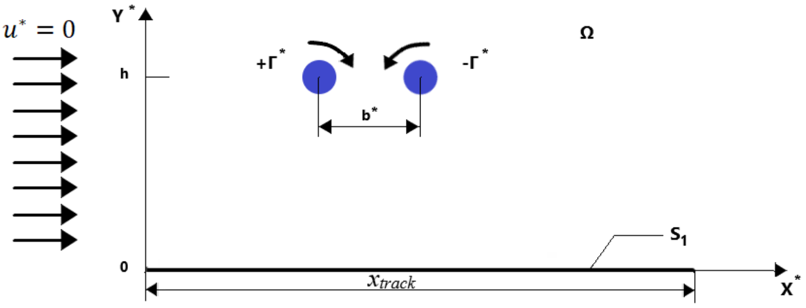


Figure 1. Model used in the numerical simulation.

The incompressible flow of a Newtonian fluid with constant properties in a two-dimensional region, Ω is governed by the continuity and the Navier-Stokes (N-S) equations, which can be written in the form:

$$\nabla^* \cdot \mathbf{u}^* = 0 \tag{2}$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla^* \mathbf{u}^* = -\frac{1}{\rho}\nabla^* p^* + \nu \nabla^{*2} \mathbf{u}^* \tag{3}$$

In Eqs. (2) and (3), u is the velocity vector field, p is the pressure field and ν is the fluid kinematic viscosity coefficient.

In order to reduce the complexity of the problem, the ground plane is considered just as an obstacle; thus, just the impermeability condition is satisfied on S_1. In accordance to this boundary condition, the normal velocity component of a fluid particle ($u_n{}^*$) must be equal to the normal velocity component of the boundary ($v_n{}^*$); see Eq. (4).

$$u_n{}^* - v_n{}^* = 0 \tag{4}$$

In order to make the non-dimensional problem, the following characteristic quantities are used:

b^* = characteristic length;
$\frac{\Gamma^*}{b^*}$ = characteristic velocity.

Thus, the continuity and the Navier-Stokes equations become:

$$\nabla \cdot \mathbf{u} = 0 \tag{5}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} \tag{6}$$

In Eq. (6), Re is the Reynolds number, which is defined as:

$$Re = \frac{\Gamma}{\upsilon} \tag{7}$$

In the same way, the impermeability condition become:

$$u_n - v_n = 0 \tag{8}$$

Taking the curl of Navier-Stokes equations, the pressure term is eliminated and the non-dimensional 2-D vorticity transport equation is obtained (Batchelor, 1967):

$$\frac{\partial\omega}{\partial t}+\mathbf{u}\cdot\nabla\omega=\frac{1}{\mathrm{Re}}\nabla^2\omega \qquad (9)$$

In Eq. (9), ω is the only non-zero component of the vorticity vector. So, $\partial\omega/\partial t$ represents the local vorticity variation rate, $\boldsymbol{u}\cdot\nabla\omega$ represents the advective vorticity variation rate and $\nabla^2\omega/Re$ is the diffusive vorticity variation rate. It is not necessary to deal with the non-linear term, $\boldsymbol{u}\cdot\nabla\omega$, because the problem is solved using a lagrangian numerical method; see more details on Section 3.

3. SOLUTION METHOD

The solution method applied in this paper uses the viscous splitting algorithm (Chorin, 1973). According to it, is possible to solve the advection and diffusion separately but in the same time increment. The advection and diffusion are governed, respectively, by:

$$\frac{\partial\omega}{\partial t}+\mathbf{u}\cdot\nabla\omega=\frac{\mathrm{D}\omega}{\mathrm{Dt}}=0 \qquad (10)$$

$$\frac{\partial\omega}{\partial t}=\frac{1}{\mathrm{Re}}\nabla^2\omega \qquad (11)$$

Advection is governed by Eq. (10), which is computed using material derivate, making evident the lagrangian characteristic of the solution method. The trajectory of a particle is governed by Eq. (12):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{dt}}=\mathbf{u}(\mathbf{x},\mathrm{t}) \qquad (12)$$

According to Eq. (12), to simulate the advection step, it is necessary to know the velocity field of the flow in the position held by each Lamb discrete vortex used to discretize the vorticity present in the fluid domain. It is possible to split the solution of advective stage in two steps: (i) Determine the velocity field at each time increment; (ii) Then, solve Eq. (12) to each discrete vortex present in the fluid domain. The computation of velocity field suffers three influences: solid boundary, in this case, the airport track ($ub(\boldsymbol{x},t)$); incident flow, but, in this research, there is no lateral wind ($u^* = 0$ in Figure 1) and the cloud of discrete vortices ($uv(\boldsymbol{x},t)$).

The contribution of solid boundary is calculated using the panel method (Katz and Plotkin, 1991). The principle of this method is to discretize the body surfaces using panels, which can be flat or curved, where singularities are disposed. These singularities can be of three types: vortex, dipole or source and constantly or linearly disposed. This paper uses source singularities which are constantly disposed on flat panels.

The solid boundary contribution is calculated by:

$$\mathrm{ub_i}^{\mathrm{n}}(\mathbf{x}_\mathrm{i},\mathrm{t})=\sum_{\mathrm{p}=1}^{\mathrm{NP}}\sigma_\mathrm{p}c_{\mathrm{ip}}^{\mathrm{n}}\left[\mathbf{x}_\mathrm{i}(\mathrm{t})-\mathbf{x}_\mathrm{p}\right],\ \mathrm{n}=1,2 \text{ and } \mathrm{i}=1,\mathrm{Z} \qquad (13)$$

Where Z is the total number of discrete vortices present in the flow at instant t, NP is the total number of source flat panels, $\sigma_p = constant$ is the source density per unit length, $c_{ip}^n\left[x_i(t)-x_p\right]$ is the $n-th$ component of the velocity induced at discrete vortex i by p panel. Note that the source strengths are determined using the impenetrability condition (Eq. 4).

Finally, the last step of advection stage when only the Biot-Savart law is used, corresponds to the calculation of the velocity field due to the vortex-vortex interaction, which requires an expensive convolution step of $O(Z^2)$ calculations, as can be seen in Eq. (14), where Γ_j is the intensity of the j vortex and $c_{ij}^n\left[x_i(t)-x_j(t)\right]$ is the $n-th$ component of the induced velocity at a discrete vortex i by a discrete vortex j.

$$\mathrm{vv_i}^{\mathrm{n}}(\mathbf{x}_\mathrm{i},\mathrm{t})=\sum_{\mathrm{j}=1}^{\mathrm{Z}}\Gamma_\mathrm{j}c_{\mathrm{ij}}^{\mathrm{n}}\left[\mathbf{x}_\mathrm{i}(\mathrm{t})-\mathbf{x}_\mathrm{j}(\mathrm{t})\right],\ \mathrm{n}=1,2 \text{ and } \mathrm{i}=1,\mathrm{Z} \qquad (14)$$

The computation of Eq. (14) represents a heavy limitation on lagrangian methods to use them to solve practical engineering problems. To reduce this limitation, the fast multipole method is implemented. This algorithm consists of clustering the influence of elements close to each other into multipole expansions and, then, evaluating their interactions at distant locations. Simulations using the fast multipole method are computed faster than the direct computation done by Biot-Savart law.

First, a pre-processing step is performed; the computational domain is divided into 4^L square boxes (L is the maximum refinement level) to cluster the influence of particles (discrete vortices) close to each other into multipole expansions. The main objective of the FMM is to promote more interactions among square boxes than interactions among particles. This feature makes the computational costs lower. Then, lists of well-separated boxes and their neighbors are determined. It is possible to create two interaction lists: considering B as the parent of a box b, B can have up to 9 boxes sharing a node, being 8 neighbors and B itself. Each one of these 9 boxes have 36 children boxes, 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. In the end of this step, the particles are mapped to determine in which box they are at the maximum refinement level.

After finishing the pre-processing step, the implementation of the fast multipole method is able to start. The first step of the implementation, called particle-to-multipole, 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 *n* discrete vortex with intensity Γ_i, with a complex distance of z_i from the center of the box (Eq. 15). The sum of all vortex particle intensities is given by Eq. (16).

$$\mathrm{a(b,k,L)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} -\Gamma_\mathrm{i} \frac{\mathrm{z_i}^\mathrm{k}}{\mathrm{k}} \tag{15}$$

$$\mathrm{Q(L)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \Gamma_\mathrm{i} \tag{16}$$

The next step, called multipole-to-multipole, is responsible to translate the influences from the center of the children box, *b*, to the center of the parent box, *B*, at level *l-1*, resulting in $a(B,k,l-1)$ - see Eq. (17). The intensity of the multipoles in level *l-1*, $Q(l-1)$, is the sum of the children's intensities, $Q(l)$ – see Eq. (18).

$$\mathrm{a(B,k,l-1)} = \sum_{\mathrm{i=1}}^{4} \left\{ \left[\sum_{\mathrm{kk=1}}^{\mathrm{k}} \mathrm{a_i(kk,l)z_i}^{\mathrm{k-kk}} \binom{\mathrm{k-1}}{\mathrm{kk-1}} \right] - \left[\mathrm{Q_i(l)} \frac{\mathrm{z_i}^\mathrm{k}}{\mathrm{k}} \right] \right\} \tag{17}$$

$$\mathrm{Q(l-1)} = \sum_{\mathrm{i=1}}^{4} \mathrm{Q_i(l)} \tag{18}$$

The step called multipole-to-multipole is made up to level 2, because this is the lowest level possible to have multipole-to-local calculations (Eq. 19). 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 $b(B,kk,l)$.

$$\mathrm{b(B,kk,l)} = \sum_{\mathrm{j=1}}^{\mathrm{nbox}} \left[\frac{\mathrm{a_j(k.l)} \left(\frac{-1}{\mathrm{z_j}}\right)^\mathrm{k} \binom{\mathrm{kk+k-1}}{\mathrm{kk-1}}}{\mathrm{z_j}^{\mathrm{kk}}} - \frac{\mathrm{Q_j(l)}}{\mathrm{kkz_j}^{\mathrm{kk}}} \right] \tag{19}$$

In Eq. (19), the variable called *nbox* represents the non-empty boxes from the interaction list of *B*, z_j is the complex distance between the box from the interaction list and the box of interest and the variables *a* and *Q* are obtained from the previously calculated multipole expansion from box *j* in the interaction list of *b(l)*.

The translation of the influence coefficients from the boxes that are in the interaction list of *B* to a child box, *b*, is made in the local-to-local step. The influence of all distant boxes of *b* is the sum of the multipole-to-local steps from the boxes of its own interaction list and also from the local-to-local steps from its parent (*B*). The Eq. (20) represents 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.

$$\mathrm{c(b,k,l+1)} = \sum_{\mathrm{i=1}}^{4} \left\{ \sum_{\mathrm{kk=k}}^{\mathrm{p}} \left[\mathrm{b(k,1)(-z_i)}^{\mathrm{kk-k}} \binom{\mathrm{kk}}{\mathrm{k}} \right] \right\} \tag{20}$$

The last step to calculate the velocity field due to the discrete vortex cloud consists of transferring the influence from the center to all particles within the box through another Taylor's series and interact the particles inside the same box and in boxes from the neighbors list through the Biot-Savart law, responsible to calculate the near field. The Eq. (21) 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*:

$$\mathrm{uv_i} = \mathrm{vv_i} + \sum_{\mathrm{k=1}}^{\mathrm{p}} \mathrm{b(k,1)kz_i}^{\mathrm{k-1}} \tag{21}$$

In Eq. (21), vv_i 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. All the calculations performed here to compute the velocity field used the lamb vortex model.

Since the velocity field is obtained, it is possible to impose a dislocation to each vortex particle. In this work, is used a first order Euler scheme. Thus:

$$\mathrm{x_i(t + \Delta t) = x_i(t) + u_i(t)\Delta t + \varsigma_i} \tag{22}$$

In Eq. (22), ς_i is the random walk displacement, which is given by (Lewis, 1999):

$$\varsigma_\mathrm{i} = \sqrt{\frac{4\Delta\mathrm{t}}{\mathrm{Re}}\ln\left(\frac{1}{\mathrm{P}}\right)}\,[\cos(2\pi\mathrm{Q}) + \mathrm{w}\sin(2\pi\mathrm{Q})] \tag{23}$$

In Eq. (23), $w = \sqrt{-1}$, *P* and *Q* are random numbers between 0 and 1.

When the core spreading method is used, there is no displacement, what means that $\varsigma_i = 0$. Instead of impose a displacement on the discrete vortices, the core spreading method stablishes an approximation of the vorticity field evolution in time, due to diffusion, imposing a growth on the discrete vortex core in accordance to Eq. (24).

$$\sigma_{0_\mathrm{i}}(\mathrm{t} + \Delta\mathrm{t}) = \sqrt{{\sigma_{0_\mathrm{i}}}^2(\mathrm{t}) + \frac{\Delta\mathrm{t}}{\mathrm{Re}}} \tag{24}$$

In Eq. (24), σ_{0_i} is the discrete vortex core and Δt is the time increment of the numerical simulation. When the vortex core reachs a maximum value, it suffers a partition process, which gives rise to four new vortices, the new particles are positioned at 90° each other and at a distance r (Eq. 25) from the original vortex.

$$\mathrm{r} = 2\sigma_{0_\mathrm{i}}\sqrt{1 - \alpha^2} \tag{25}$$

In Eq. (25), $\alpha[0,1]$ is a numerical parameter that governs the partition process (Rossi, 1996).

4. RESULTS

As already been mentioned, in order to simplify the numerical implementation, the simulations presented here are performed considering the solid boundary only as an obstacle, so, the vorticity generation is not taken into account; besides that, the solution of the vorticity diffusion equation (Eq. 11) is obtained using the random walk method and the core spreading method. The purpose is to compare the accuracy of the results obtained by both diffusion equation solutions and to investigate the CPU time reduction provided by the fast multipole method (FMM). Thus, two identical vortex clouds positioned as shown in Figure 1 are used to simulate the vortices detached from airplane wings during a landing or a take-off procedure. Each vortex cloud is composed by *nz* discrete vortices with intensity $\pm 1/nz$ and the ground was discretized in *NP* flat panels.

For the creation of the clouds, the *nz* discrete vortices are initially concentrated on the origin of the reference system; then, they are submitted to a diffusion dislocation using random numbers until the more distant discrete vortex reach a distance equal to $0.1b^*$ (see Figure 1). This way, one ensures a gaussian distribution of the vorticity contained in the clouds as if it were a big lamb vortex. The Euler first order time marching scheme is used in all simulations.

First, a simulation using the random walk method is performed. The velocity field is computed using two different approaches: (i) only through the Biot-Savart law; (ii) and only through the fast multipole method, increasing the refinement levels according the number of particles inside the boxes, starting from the maximum refinement level equal to 4 and, then, when the ratio between the number of particle in the fluid domain and the amount of boxes in the maximum refinement level exceeds 197, the domain is refined; the maximum refinement level using in this work is *L* = 6 (Vidille, 2022). Especially in this case, whereby the amount of particles in the fluid domain is stationary, because the diffusion is simulated through the random walk method and the vorticity generation on the track is not considered, the maximum refinement level does not change until the end of the simulation.

The numerical parameters used in this simulation are: $Re = 7.65 \times 10^3$, $nz = 1.0 \times 10^2$, $NP = 240$, $\Delta t = 0.05$; the simulations are performed until $t = 20$. In the simulations whereby the FMM algorithm is considered, the number of terms used on the Taylor series (p) is equal to 25. The computational time of the simulations using the random walk method is shown in Tab.1. The numerical errors (E) according to the dimensionless time (t) for those simulations are presented in Figure 2. The amount of discrete vortices in the end of the simulations, Z, is equal to 200 in these two simulations.

Table 1. Computational time of the simulations performed with the diffusion simulated through the random walk method.

Cases	Computational time (s)	Method used to compute the velocity field
1	8	BS
2	47	FMM

All the numerical errors presented in this paper, calculated in order to evaluate the accuracy of the FMM during the simulations, are computed using Eq. (26), as done by Carrier *et al.* (1988):

$$E = \left(\frac{\sum_{i=1}^{Z} \left(u_{FMM_i} - u_{BS_i} \right)^2}{\sum_{i=1}^{Z} u_{BS_i}^2} \right)^{1/2} \tag{26}$$

In Eq. (26), E is the error, u_{FMM_i} is the velocity field computed in discrete vortex *i* position through the FMM method and u_{BS_i} is the velocity field computed in discrete vortex *i* position through the Biot-Savart law.

The computational time, in the simulation performed using the random walk method, increases with the use of the FMM algorithm. This behavior is explained because the amount of particles in the fluid domain is too small and does not justify the use of the fast multipole method to those simulations, because the cost to perform the preprocessing steps of the FMM overcomes the intrinsic cost of the Biot-Savart law. The numerical error, in the end of the simulation is 38.6% (Figure 2). This value is very high, what confirms that the use of the fast multipole method, in this case, is not recommended. Therefore, the use of FMM algorithm in cases where the amount of particles in the fluid domain is low, does not have any advantage.

When the diffusion equation is solved by the core spreading method, the use of fast multipole method is particular important, since the number of particles increases a lot. In such situation the velocity field is computed using two approaches: (i) only through the Biot-Savart law; (ii) through the FMM, after the number of particles in the computational domain exceeds 50000, and increasing the refinement level in accordance to the criterion previously described. This strategy allows an appropriate comparison between the Biot-Savart law algorithm and the fast multipole algorithm, because, according to Figure (2), the numerical error in the velocity field, when the fast multipole method is used, increases during the simulation. So, beginning the simulation using the Biot-Savart law and turning on the FMM algorithm just when the number of particles is large enough, the FMM algorithm is optimized and, as consequence, makes the numerical error lower.

The simulations performed with the core spreading method uses the same parameters as the ones performed with the random walk method. The parameter α, that governs the partition process, is choosed as $\alpha = 0.2$ and $\alpha = 0.3$. The computational time of the simulations using the core spreading method is shown in Tab.2. The numerical errors (E) according to the dimensionless time (t) for these simulations are presented in Figure 3.

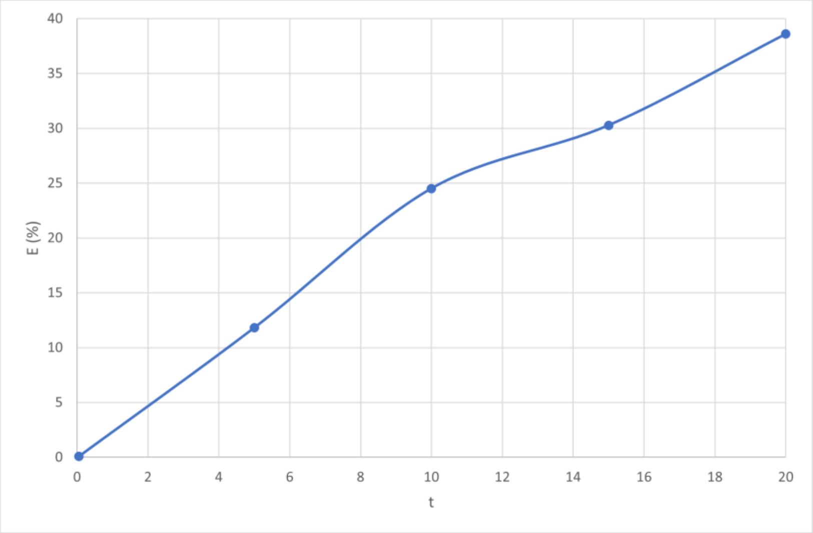


Figure 2. Accumulated error to compute the velocity field using the fast multipole method for the simulations whereby the diffusion is computed through the random walk method (Euler; $\Delta t = 0.05$; $p = 25$; $nz = 1.0 \times 10^2$; $Re = 7.65 \times 10^3$).

Table 2. Computational time of the simulations performed with the diffusion simulated through the core spreading method.

Cases	Computational time (h)	Method used to compute the velocity field	α
3	22.1	BS	0.2
4	26.6	BS	0.3
5	2.1	FMM	0.2
6	2.6	FMM	0.3

In these two simulations, the computational time was reduced with the use of the fast multipole method. In the simulation where $\alpha = 0.2$, the reduction in the CPU time is 90.5% and $Z = 204800$; in the simulation where $\alpha = 0.3$, the reduction in the CPU time is 90.2% and $Z = 819200$.

It is known that the numerical errors of the fast multipole method accumulate over time (Figure 2). Thus, since the FMM algorithm is activated earlier for α = 0.3 (the computational domain reachs 50000 particles faster for α = 0.3 than for α = 0.2), its error is higher (Figure 3).

The numerical errors, in the end of the simulations, are 2.08% and 2.53% (Figure 3), for $\alpha = 0.2$ and $\alpha = 0.3$, respectively. The computational time reduction and acceptable numerical errors show that the fast multipole method was correctly implemented and is recommended to situations whereby the amount of particles in the fluid domain is considerably high.

In Figure (4) is shown the trajectories of right side vortex cloud. The numerical results obtained are compared to the experimental conducted by Zheng and Ash[1] (1996, apud Liu and Srnsky, 1990). It is important to keep in mind that the simulations are performed in a way that the solid boundary represent just an obstacle; just the impermeability condition is satisfied on that boundary, no vorticity is generated on it and, as consequence, it is not fair to expect a fidelity reproduction of the experimental results, especially the rise and the winding of vortices. Even though, some important observations can be obtained.

It is possible to observe, in Figure 4(a), how the FMM algorithm presents bad results when the number of particles in the computational domain is not great enough. On the other hand, when the core spreading method is used, the number of particles in the domain increases a lot and, as consequence, the curves of Biot-Savart law computation and FMM computation are overlapped. Finally, one can observe from Figure 4(c) that the core spreading method (with α = 0,3) predicts the trajectory closest to the experimental result (except for rising and winding), which indicates how promising is the core spreading method, especially for higher values of α.

[1] Liu, H.T. and Srnsky, R.A., 1990. “Laboratory investigation of atmospheric effects on vortex wakes”. *Flow Research, Inc.*.

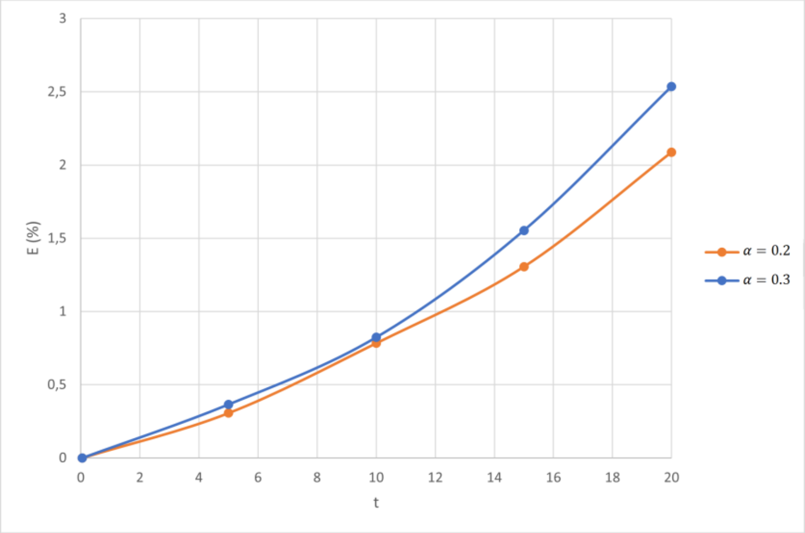
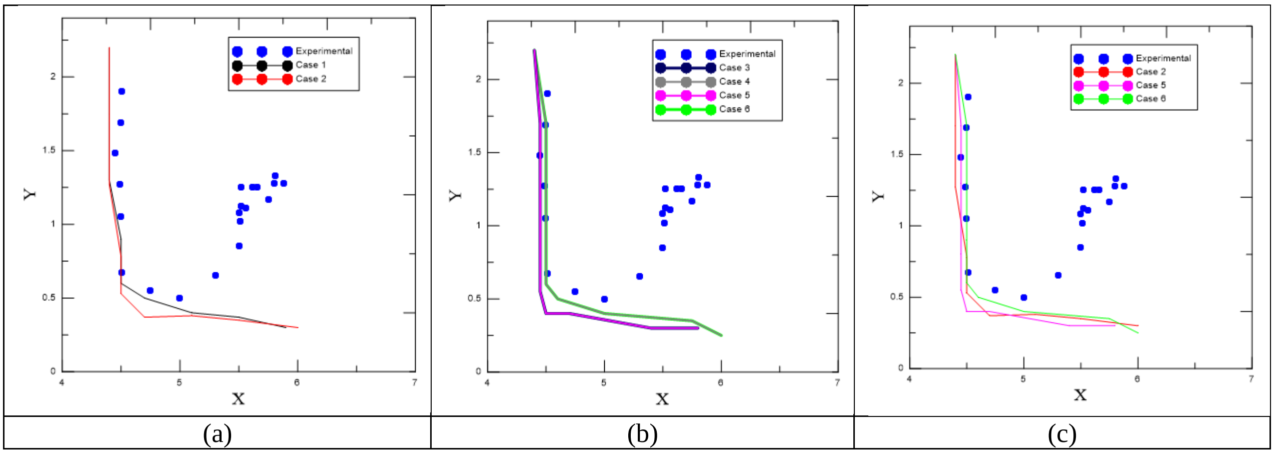


Figure 3. Accumulated error to compute the velocity field using the fast multipole method for the simulations whereby the diffusion is computed through the core spreading method (Euler; $\Delta t = 0.05$; $p = 25$; $nz = 1.0 \times 10^2$; $Re = 7.65 \times 10^3$).



(a) (b) (c)

Figure 4. Comparison between the experimental results and the simulations presented of the right side of the discrete vortices cloud.

5. CONCLUSIONS

An algorithm of the fast multipole method based on the work developed by Ricciardi (2016) was implemented to study the possibility to make feasible the use of the modified core spreading method to solve the vorticity diffusion equation. The first analysis about the behavior of the numerical error shows that in situations whereby the amount of particles in the fluid domain is not significant, the use of the FMM algorithm is not recommended, because it is not able to reduce the CPU time and the numerical error of the velocity field becomes unreasonable. Then, it was decided to evaluate how the FMM algorithm behaves in simulations with significative number of particles in the fluid domain, performed with the diffusion calculated through the core spreading method. With this approach, it was possible to note that the FMM algorithm was successfully implemented since the CPU time decreased at least 90% in comparison with the simulations whereby the velocity field due to the vortex-vortex interaction was calculated only through the Biot-Savart law. Besides that, the numerical error, at the end of the simulations was, in the maximus, equal to 2.53%. According to the results presented, the core spreading method improves the results of the discrete vortices cloud behavior, when it is used with higher values of α.

It is possible, from the present work, to develop researches in order to evaluate the behavior of the counterrotating pair of vortices detached from airplane wings using the core spreading method, specially using higher values of the parameter α, to calculate the diffusion and considering the vorticity generation on the solid boundary, in order to make a comparison between the numerical results and the experimental ones. To make these simulations feasible, it is necessary to use the FMM algorithm and implement two parameters: the coalescence factor and the minimum intensity of the discrete vortices, which determines the minimum value of intensity of the discrete vortices able to suffer the process of partition in the core spreading method. This tool is used to avoid that a discrete vortex, which has a tiny intensity, splits. The coalescence of the discrete vortices consists in grouping the particles physically close, transforming a cluster of discrete vortices into a single discrete vortex, keeping the momentum and circulation, which decreases the amount of particles in the fluid domain.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge São Paulo Research Foundation (FAPESP), grant 2022/03630-4, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), grant 88887.690941/2022-00.

7. REFERENCES

Alcântara Pereira, L.A., Hirata, M.H. and Manzanares Filho, N., 2004. "Wake and Aerodynamics Loads in Multiple Bodies - Application to Turbomachinery Blade Rows". *J. Wind Eng. Ind Aerodyn*., Vol. 92, pp. 477-491.

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

Bimbato, A.M., Alcântara Pereira, L.A. and Hirata, M.H., 2018. "Development of a new Lagrangian vortex method for evaluating effects of surfaces roughness". *European Journal of Mechanics B-Fluids*, Vol. 74, pp. 291-301.

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..

Leonard, A., 1980, "Vortex methods for flow simulation". *Journal of Computational Physics*, Vol. 37, pp. 289-335.

Lewis, R.I., 1991. *Vortex element method for fluid dynamic analysis of engineering systems.* Cambridge Univ. Press, Cambridge, England, U.K.

Lewis, R.I., 1999. "Vortex Element Methods, the Most Natural Approach to Flow Simulation - A Review of Methodology with Applications". In *Proceedings of 1st Int. Conference on Vortex Methods*. Kobe, Japan.

Machol, R., 1993. "Wake Vortices – A Primer*". FAA Aviation Safety Journal*, Vol. 3, No. 1, pp. 16-19.

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

Ricciardi, T.R., 2016. *Fast multipole discrete vortex method applied to unsteady flow simulations*. M.Sc. Dissertation, University of Campinas.

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.

Rossi, L.F., 1996. "Resurrecting Core Spreading Vortex Methods: A New Scheme that is both Deterministic and Convergent". *SIAM J. Sci. Comput*., Vol. 17, No. 2, pp. 370-397.

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., 2022. *Utilização de um algoritmo acelerador para simular o desenvolvimento de esteiras de aeronaves através de um método de partículas em conjunto com um método de elementos de contorno*. Undergraduate Thesis, São Paulo State University.

Zheng, Z.C. and Ash, R.L., 1996. "Study of Aircraft Wake Vortex Behavior Near the Ground". *AIAA Journal*, Vol. 34, No. 3, pp. 580-588.

8. RESPONSIBILITY NOTICE

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