FLAT PLATE BOUNDARY LAYER SIMULATION USING THE VORTEX METHOD

Victor Santoro Santiago
vsantoro@ig.com.br

Gustavo C. R. Bodstein
Universidade Federal do Rio de Janeiro, Departamento de Engenharia Mecânica – Poli/COPPE
Centro de Tecnologia, Bloco G, sala 204 – Ilha do Fundão, 21945-970 Rio de Janeiro, RJ – Brasil
gustavo@mecanica.coppe.ufrj.br

Abstract. *The Vortex Method has been used to simulate a large number of external flows around bodies. Despite the tremendous improvement achieved during the last decades, this powerful mesh-free technique requires special attention to model the vorticity generation process, the lagrangian transport of vorticity by diffusion, and the high computational cost of the lagrangian transport of vorticity by convection. This work proposes a new and efficient two-dimensional algorithm that improves the vorticity generation and the diffusion models. In addition, an Adaptive Multipole Expansion code is employed to calculate the induced velocities of the vortices, which promotes an enormous decrease in the convective computational effort. Lamb vortices are generated near the wall by diffusion, and the Corrected Core-Spreading Method is used to simulate the vorticity diffusion in the boundary layer and wake. The convective motion of the vortex cloud is calculated using the Adams-Bashforth second-order time-marching scheme. The algorithm is tested for the well-known problem of two-dimensional, incompressible flow over a flat plate. The agreement with the flat-plate Blasius solution indicates that the algorithm furnishes a good representation of the generation, diffusion and convection processes of vorticity in the flow.*

Keywords*: Vortex method, corrected core-spreading method, adaptive fast multipole method, flat-plate boundary layer, Blasius boundary layer*

1. INTRODUCTION

This paper describes a simulation of the flow over a flat plate using the Vortex Method. Some improvements on the vortex method proposed by Santiago and Bodstein (2006) are here implemented and applied to the well-known Blasius boundary layer flow problem. This benchmark problem is chosen because of the existence of an exact solution that can be used for comparison. The flat plate boundary layer flow problem is perfect for the evaluation of the convective-diffusive vorticity transport mechanisms that are carried out at during the simulation. In addition, we pay special attention to the vorticity creation mechanism at the wall and its subsequent shedding into the flow field. This step of the algorithm is responsible for the transfer of the exact amount of the vorticity generated at the wall into the fluid by diffusion and also for the determination of the adequate position to place the nascent vortices within the fluid, in the neighborhood of the plate. The simulation of the Blasius flat plate problem allows for a detailed assessment of the new numerical schemes that have been developed within the vortex method framework to simulate high Reynolds number flows.

The basic idea of Vortex Method relies on the discretization of the vorticity field into a cloud of free vortex blobs to simulate the convective-diffusive transport of vorticity in the flow. Discrete vortex blobs are generated in the neighborhood of the solid wall in order to satisfy the no-slip and the no-penetration boundary conditions, and they move in a Lagrangian manner to satisfy the vorticity transport equation. In this process, advantage is taken of the property that the boundary condition at infinity is automatically satisfied due to the decay of the basis functions used to model the vorticity field of these vortices. In order to determine the evolution of the flow filed, these vortex blobs move according to vorticity-velocity relations. The Lamb vortex model is employed to avoid the singular behavior of point vortices at the origin, which introduces a core for each discrete vortex.

The diffusion process of vorticity is simulated employing the Corrected Core-Spreading Method (Rossi, 1996). In this methodology, the vortex core radius grows in time from a minimum value up to a maximum pre-determined value. At this point, the vortex splits into four new vortices with smaller core radius and strength, replacing the original vortex. The newly born vortices are positioned in the neighborhood of the original vortex, such that they maintain overlapping of their core radius. This scheme is accurate and converges fast, as long as core overlapping is maintained at all times. On the other hand, this procedure imposes an outstanding computational effort because the number of vortices in the cloud grows exponentially in time. To deal with this problem, the merging and cut-off algorithms proposed by Rossi (1997) are implemented. The first algorithm merges vortices that have their radius in excessive overlapping, but leaving sufficient overlapping needed to ensure convergence of the Vortex Method. The second algorithm provides a lower bound for the value of the vortex strength that can be reached by the vortices, preventing vortices from splitting during the diffusion process when their circulation become smaller than a specified value.

A high computational cost is the main problem of the convective vorticity transport modeled by the Lagrangian vortex motion, since the velocity field induced by the vortices on each other has to be evaluated at the positions of all vortices. This vortex-vortex interaction has an operation count of the order of N_v^2, where N_v is the total number of vortices. For this reason, the Adaptive Fast Multipole Method, developed by Carrier *et al*. (1988), is used to accelerate the induced velocity calculation. This scheme reduces the operation count down to the order of N_v.

Figure 1 shows a schematic diagram of the numerical algorithm that we have implemented. The flat plate discretization step is responsible for setting up N elements on the plate and for determining the coordinates of the center of each element, called control points, where the boundary conditions are imposed. Next, the circulation of each new vortex blob to be created is determined. The positions in the neighborhood of the plate where these new vortex blobs are placed are then calculated such that the no-slip boundary condition is satisfied. We show that this position has great influence on the results of the simulation, since it contributes to impose a zero-velocity value at the wall when evaluating the velocity field. Vortices that cross the plate during either the convective step or the vortex splitting process of the diffusion step are reflected back into the flow and placed at the corresponding symmetric position. Merging of vortices with excessive overlapping and cut-off of the splitting process when the vortex strength decreases below a specified value are numerical techniques employed to limit the computational effort.

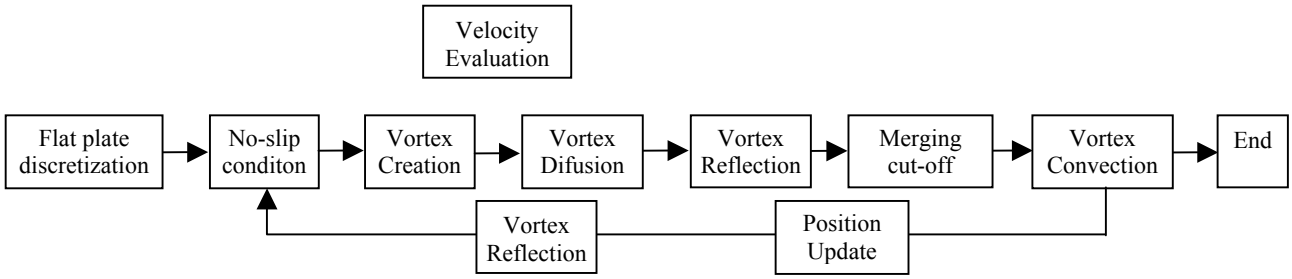


Figure1. Simplified schematic diagram of the numerical algorithm.

2. MATHEMATICAL FORMULATION

The flat plate boundary layer flow develops from an impulsive start of a uniform flow U over a two-dimensional flat plate aligned with the flow and discretized into N elements, where the no-slip and the no-penetration boundary conditions are satisfied at the control points. The induced velocity field at each control point is the result of the superposition of the uniform potential flow U and the cloud of free vortices present in the flow. The reflection method (Lewis, 1991) is applied to ensure that the no-penetration boundary condition is satisfied. In other words, if M vortices are generated above the plate, then M image vortices with opposite strength are also generated below the plate at the corresponding mirror-image position.

The vortex blobs have a vorticity basis function that is modeled using the Lamb vortex model, and the vorticity-velocity relation for the induced velocity field $\mathbf{q}$ may be written as

$$\mathbf{q}(r,t) = -\frac{\Gamma}{2\pi r}\left[1-\exp\left(-\frac{r^2}{\sigma^2}\right)\right], \tag{1}$$

where Γ is the vortex circulation (or strength), σ is the core radius that desingularizes the blob and r is the distance from the vortex location to the point where the induced velocity is calculated.

2.1. No-slip boundary condition

The velocity component in the x-direction evaluated at the i^{th}-control-point of the plate due to the uniform incident flow with freestream speed U and all M free vortex blobs in the cloud and their images is given by

$$\gamma_i = U - \frac{1}{\pi}\sum_{j=1}^{M}\frac{\Delta\Gamma_j y_j}{r^2}\left[1-\exp\left(-\frac{r^2}{\sigma^2}\right)\right]. \tag{2}$$

Equation (2) determines the vorticity per unit length $\gamma(s_i)$ of each plate element i of length Δs_i to be shed into the flow in the form of L nascent vortices with circulation $\gamma(s_i)\Delta S_i/L$. This amount of vorticity distributed over the L new vortices is created in the flow to cancel exactly the velocity at the plate and, therefore, to satisfy the no-slip condition.

2.2. Vortex creation

After evaluating the vorticity γ_i from Eq. (2), the position of new vortices is established such that the no-slip condition is satisfied. It is desirable to choose the position of the center of the newly created vortex and its displacement ε from the flat plate in such a way that the no-slip condition continues to be valid at the beginning of next step. This is an improvement when compared to the classical procedure that sets $\varepsilon = \sigma$, but the problem leads to an unusual non-linear system of algebraic equations. It is possible to employ an alternative procedure to calculate ε_i for control-point i such that the no-slip boundary condition remains valid. To avoid the non-linear system of equation, we approximate all ε_j, $j \neq i$, by their values at the previous time step. The unknown ε_i is then calculated explicitly and an iterative procedure is implemented for all ε_j, $j \neq i$, until convergence to a specified value is reached. Figure 2 shows the vortex position during the vortex creation process.

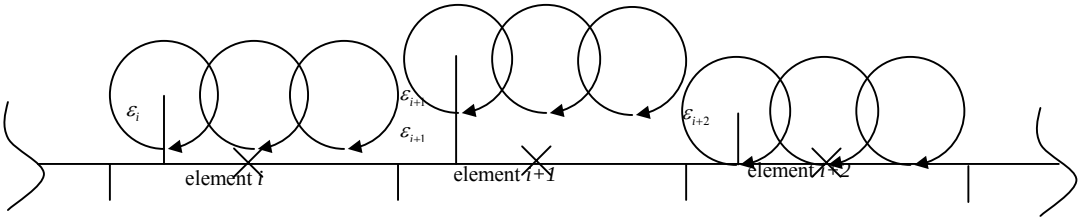


Figure 2. Creation of L vortices for all N control points.

The approximation for ε_j, $j \neq i$, considering the influence of all new $M = LN$ vortices, including their images, leads to the following equation

$$\gamma_i - \frac{\gamma_i \Delta S_i}{\pi L \varepsilon_i} - \sum_{\substack{j=1 \\ j \neq i}}^{LN} \frac{\gamma_j \Delta S_j \varepsilon_j^{t-1}}{\pi L r_{ji}^{\,2}} = 0 \,, \tag{3}$$

where Eq. (3) is solved for ε_i; the value of ε_j^{t-1} is obtained from the previous time step and γ_i is calculated from Eq. (2).

Figure 3, generated using the results from simulations with and without the use of the model above, shows how the use of Eq. (3) can improve the evaluation of velocity field near the wall. The simulation obtained by setting $\varepsilon = \sigma$ is very poor compared to the simulation that evaluates ε_i from Eq. (3). This new model is more accurate to produce both good force calculation on the surface and good development of the simulation as a whole.

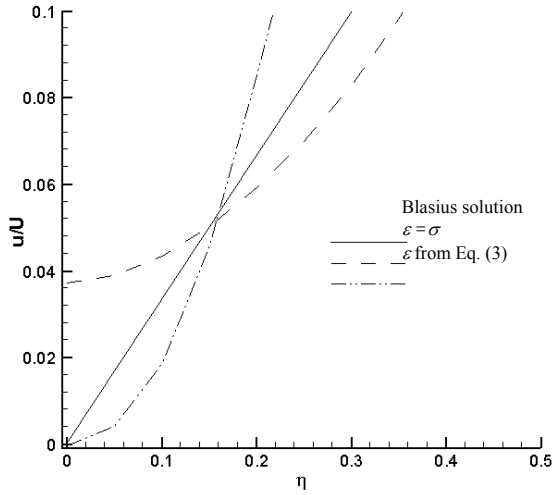


Figure 3. Comparison between the velocity field near the wall generated with $\varepsilon = \sigma$ and ε calculated for Eq. (3).

The diffusion layer that develops during the vorticity generation process at the wall, shown in Fig. 4, has an area equal to the product of the Rayleigh diffusion displacement thickness, $\delta_v = 2,76\sqrt{\Delta t/Re}$, to the length of element *i*, ΔS_i . This diffusion area and the number of vortices *L* created per plate element are used to calculate the radius σ of the nascent vortices according to

$$\sigma = \sqrt{\frac{\Delta S_i \delta_v}{\pi L}} . \tag{4}$$

The number of vortices *L* created per plate element is the integer number obtained from $L = \mathrm{INT}(\Delta S_i/\delta_v) + 1$, where $\Delta S_i/\delta_v$ is the aspect ratio of the diffusion layer. Since ΔS_i and δ_v are kept constant during the simulation, *L* is the same for all plate elements during all time steps. The circulation of each new (nascent) vortex is calculated according to

$$\Gamma_k = \frac{\gamma_i \Delta S_i}{L}, \; 1 \le k \le .\, L. \tag{5}$$

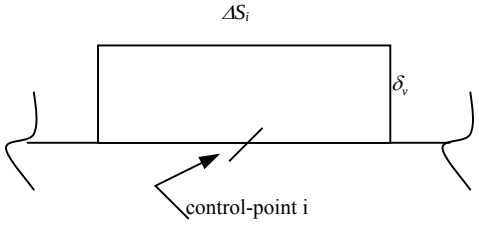


Figure 4. The diffusion area originating from the vorticity creation at plate element *i*.

2.3. Field evaluation

The velocity field is evaluated using the vorticity-velocity relation obtained from the Biot-Savart law and the Lamb vortex model described by Eq. (1). Because we use the reflection method (Lewis, 1991) to impose the no-penetration boundary condition on the plate, our formulation includes the image vortices as well. Therefore, the *x* and *y* components of the velocity vector **q** induced by the i^{th}-vortex at the j^{th}-domain point, summed over all the N_v vortices in the cloud and their images, are given by

$$\Delta u_j = \sum_{i=1}^{N_v} \frac{\Delta\Gamma_i}{2\pi}\left[\frac{y_j - y_i}{r_1^2}\left\{1 - \exp\left(-5,02572\frac{r_1^2}{\sigma_i^2}\right)\right\} - \frac{y_j + y_i}{r_2^2}\left\{1 - \exp\left(-5,02572\frac{r_2^2}{\sigma_i^2}\right)\right\}\right], \tag{6a}$$

$$\Delta v_j = \sum_{i=1}^{N_v} \frac{\Delta\Gamma_i}{2\pi}\left[\frac{x_j - x_i}{r_2^2}\left\{1 - \exp\left(-5,02572\frac{r_2^2}{\sigma_i^2}\right)\right\} - \frac{x_j - x_i}{r_1^2}\left\{1 - \exp\left(-5,02572\frac{r_1^2}{\sigma_i^2}\right)\right\}\right], \tag{6b}$$

where $r_1^2 = \left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2$, $r_2^2 = \left(x_j - x_i\right)^2 + \left(y_j + y_i\right)^2$ and $\sigma_i = 2r_{\max}$. The tangential velocity is a maximum when the distance from the vortex core center is $r_{\max}$. When the j^{th}-point is the i^{th}-vortex, the first term in curly brackets in Eqs. (6) is replaced by the following equations

$$\Delta u_j = -\frac{\Delta\Gamma_j}{4\pi y_j}\left\{1 - \exp\left(-5,02572\frac{2y_j^2}{\sigma_j^2}\right)\right\}, \tag{7a}$$

$$\Delta v_j = 0 . \tag{7b}$$

It is necessary to correct the vorticity field when the vortex core crosses the flat plate, as shown in Fig. 5, because part of the vorticity is now inside the wall and it does not contribute to the vorticity and velocity fields. This correction is implemented by modifying the vortex circulation according to

$$\Gamma = \int_{-\theta_1}^{\theta_2} \omega dA = \int_{-\theta_1}^{\theta_2}\int_0^{\infty} \frac{\Gamma^*}{\pi\sigma^2}\exp\left(-\frac{r^2}{\sigma^2}\right) r\, dr\, d\theta = \left(1 - \phi/2\pi\right)\Gamma^*, \tag{8}$$

where θ_1, θ_2 and ϕ are shown in Fig. 5. Consequently, the vortex core radius is also corrected by the equation

$$\sigma = \sqrt{1 - \frac{(\phi - \text{sen}\phi)}{2\pi}}\sigma^* . \tag{9}$$

Equation (9) is derived from the relation $A^* = A_T - A^{**}$, where A_T the total area of the vortex, A^* is the area above the flat plate and A^{**} is the area below the flat plate.

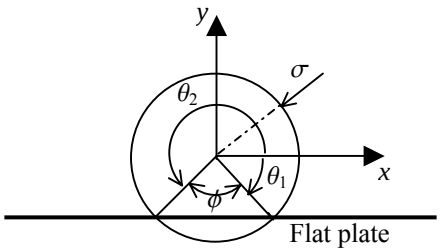


Figure 5. Vortex core radius crossing the flat plate.

2.3. Vortex diffusion

In the vortex method the diffusive transport of vorticity is related directly to how the vorticity created at the wall is shed and diffused into the flow. The first paper to attack the problem of vorticity diffusion in Lagrangian computational fluid dynamics simulations is due to Chorin (1973), who proposes a stochastic method, called the Random Walk Method, based on random displacements of the vortices. This method was considered for some time the only way to deal with vorticity diffusion in a Lagrangian description of the flow. In the last couple of decades, many deterministic models have been developed (Santiago and Bodstein, 2006). One of the most popular is the original Core-Spreading Method (Leonard, 1980), which allows the vortex core radius to grow in time according to a diffusion rate that is a solution of the vorticity diffusion equation. This method was proved by Greengard (1985) to diverge from the Navier-Stokes equation. The Corrected Core-Spreading Method (CCSM), due to Rossi (1996), proposes the splitting of the vortex into four new vortices as soon as the radius reaches a maximum value. This modification introduced by Rossi has turned the method convergent and, therefore, it has become again a usable diffusion technique. This method provides accurate deterministic results and it is very simple to implement numerically.

The basic idea of CCSM is to let the vortex core radius grow in time from a minimum value, σ_{min}, to a maximum value, σ_{max}, simulating the vortex diffusion process. This growth is controlled by a numerical parameter α, which has values in the range [0,1], where σ_{max} is input to the algorithm and $\sigma_{min} = \alpha\sigma_{max}$. The splitting process occurs when the vortex reaches a radius greater than σ_{max}. So every time σ becomes greater than σ_{max}, the code splits the vortex in four new vortices with circulation $\Gamma/4$, uniformly placed at a distance r from the original vortex calculated to conserve the vorticity second-order moment. The radius' growth rate and the distance r are given by

$$\frac{\partial \sigma^2}{\partial t} = \nu \text{ ,} \tag{10}$$

$$r = 2\sigma\sqrt{1-\alpha^2} \text{ ,} \tag{11}$$

where ν is the kinematic viscosity.

2.4. Vortex Reflection

As a result of the refinement of the vorticity field due to diffusion and of the time discretization of the convective step, a vortex may end up at a position below the flat plate. This may occur after the diffusion step, when a new vortex is created below the plate from the splitting process, or after the vortex convection, when the vortex displacement calculated with the Adams-Bashforth second-order time-marching scheme places the vortex below the wall. Hence, it is necessary to call the reflection subroutine after these both steps, as it can be seen in the Fig. 1.

2.5. Merging and cut-off

Compared with the Random Walk Method (Chorin, 1973), CCSM increases the accuracy because it is a deterministic method that discretizes the vorticity field by splitting the vortices in the cloud. However, this characteristic increases the computational cost of CCSM unboundedly, since the number of vortices increases exponentially. This problem can be dealt with using Rossi's (1997) merging and a cut-off schemes.

Merging vortices is the action that substitutes a set of overlapping vortices by one only. This procedure approximates the field generated by this vortex but it keeps the approximation error under control. In this paper, we use the algorithm proposed by Rossi (1997), where the vorticity field is approximated by a linear combination of N basic functions, as follows

$$\omega(\mathbf{x}) = \sum_{i=1}^{N} \frac{\Gamma_i}{4\pi\sigma_i^2} \exp\left[-\frac{|\mathbf{x} - \mathbf{x}_i|^2}{4\sigma_i^2} \right], \tag{12}$$

where each basic function is defined by three parameters (Γ_i, $\mathbf{x}_i$, σ_i), corresponding to the vorticity, the position and radius of the vortex i, respectively.

A maximum and minimum discretization resolution must be established to start the algorithm. This is done choosing $l_m \leq \sigma_i \leq l_M$ to obtain good vortex radius overlapping and avoid regions with insufficient discretization. In our algorithm, we set $l_M = \sigma_{\max}$ and $l_m = \sigma_{\min}$.

After merging n vortices, the merging error is

$$e(\mathbf{x}) = \frac{\Gamma_0}{4\pi\sigma_0^2} \left[\exp\left(-\frac{|\mathbf{x}|^2}{4\sigma_0^2} \right) - \sum_{i=1}^{n} \frac{\Gamma_i}{\gamma_0} \frac{\sigma_0^2}{\sigma_i^2} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_i|^2}{4\sigma_i^2} \right) \right]. \tag{13}$$

The substitution of a set of vortices by one vortex is carried out such that the zeroth, first and second moments of vorticity are maintained according to the following equations

$$\Gamma_0 = \sum_{i=1}^{n} \Gamma_i \,, \qquad \Gamma_0 \mathbf{x}_0 = \sum_{i=1}^{n} \Gamma_i \mathbf{x}_i \qquad \text{and} \qquad 4\Gamma_0 \sigma_0^2 = \sum_{i=1}^{n} \Gamma_i \left(4\sigma_i^2 + |\mathbf{x}_i - \mathbf{x}_0|^2 \right). \tag{14a, b, c}$$

Rossi's (1997) algorithm is written to identify a set of n vortices according to Eqs. (14a, b, c), bounding the error by Eq. (13). With this methodology, the error is bounded in space and the computational vorticity field only experiences controllably small instantaneous perturbations. Thus, the largest error in the entire simulation is similar to the largest local error.

The cut-off scheme is another way to control the problem size, *i.e.*, the total number of vortices in the cloud. The cut-off scheme interrupts the splitting process of vortices with sufficiently small circulation. This is a mechanism that limits the discretization of the vorticity field and helps to prevent the exponential growth of the number of vortices. More details can be seen in Rossi (1997). In the simulation presented in this work the cut-off value was chosen to be 1.0×10^{-6}.

2.6. Vortex convection

The direct evaluation of the convective velocity of all N_v vortices is a pair-wise vortex-vortex interaction, which has computational cost of the order of N_v^2. Alternatively to the direct vortex-vortex calculation, this work uses the Adaptive Fast Multipole Method (AFMM), developed by Carrier *et al*. (1988), to calculate the velocity of each vortex. This acceleration procedure has the ability to reduce the computational effort down to order N_v, dividing the vortex cloud into boxes that are considered far enough apart by a numerical parameter. The velocity induced by one box at the others is calculated expanding the Biot Savart law into a Laurent series. The number of terms in the Laurent series expansion is determined for a desired accuracy, and the domain is divided into smaller and smaller boxes until each box contain a maximum number of vortices per box; at this point, the spatial refinement stops. The interactions are carried out from one box to all other boxes, and vortices inside the same box are subject to direct vortex-vortex calculation.

Santiago *et al*. (2006) have performed a series of tests with the algorithm of the AFMM and observed that, when the vortex density in the wake becomes very high, the smallest boxes become of the order of the vortex core radius. In this case, the algorithm loses performance instantaneously and the maximum number of vortices per box must be increased. As soon as the maximum number of vortices per box is increased, the algorithm performance returns to its original computational cost of order N_v.

Another particular aspect of the AFMM of Carrier's *et al*. (1988) is that the near approach distance for desingularization can be implemented to work together with the main routine. This is equivalent to saying that the algorithm can use the Lamb vortex model, but the size of the smallest box has to be greater than the radius of the vortices, because all vortices inside the same box has to undergo direct calculation, as explained above.

The Adaptive Fast Multipole Algorithm is a very useful tool that allows longer simulations to be executed with better discretization of the vorticity field, since it makes possible to work with larger wakes (larger problem sizes). Too big problem sizes are prohibitive if using the direct calculation method based on the Biot-Savart law.

Because we use the reflection method to impose the no-penetration boundary condition, it becomes necessary to input the image cloud into the AFMM routine together with the physical vortex cloud. Although the convective step works with $2N_v$ vortices, it is still advantageous, since the reduction in the CPU time offered by N_v–computational cost overcomes the increase in the problem size.

2.7. Vortex Position Update

The step that updates the position of each vortex in the cloud is also responsible for the determination of their total displacement. This calculation is performed at the end of time step because it is necessary to know beforehand the convective velocity calculated with the algorithm of the AFMM. Knowing the velocity of all vortices in the cloud, we employ the Adams-Bashforth second-order time-marching scheme to update the position of the vortex cloud, which requires the information of the velocity field calculated at the current and the previous time steps. So, we use the first-order Euler scheme in the first step of the simulation and for all the nascent vortices, and a second-order Adams-Bashforth scheme for the remaining vortices of the cloud at any time step. The *x* and *y* components of the displacements calculated by the first-order Euler scheme and the second-order Adams-Bashforth scheme, respectively, are given by

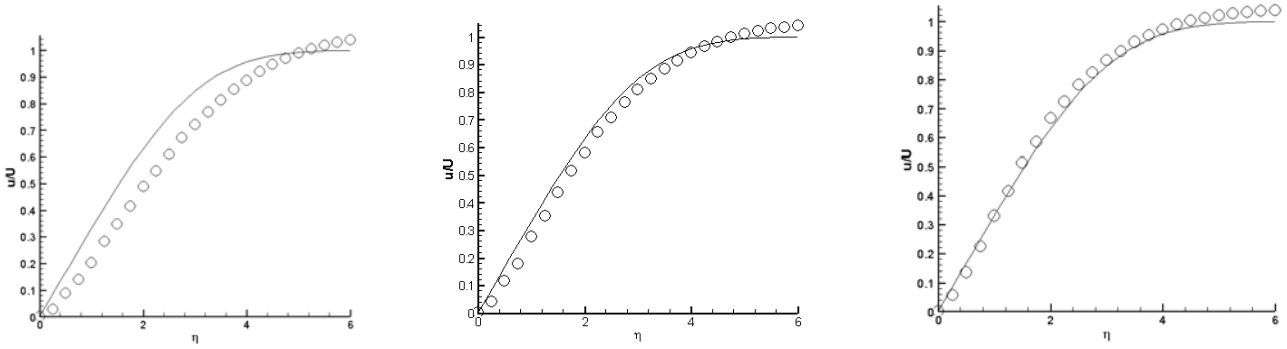
$$\Delta x_C = u(t)\Delta t \quad \text{and} \quad \Delta y_C = v(t)\Delta t \,, \tag{15a, b}$$

$$\Delta x_C = \left[1,5u(t) - 0,5u(t-\Delta t)\right]\Delta t \quad \text{and} \quad \Delta y_C = \left[1,5v(t) - 0,5v(t-\Delta t)\right]\Delta t \,. \tag{16,a, b}$$

3. RESULTS AND DISCUSSION

A good simulation using the vortex method is essentially related to a good representation of the vorticity field everywhere in the rotational flow region. In particular, the vorticity and its flux at the solid surface calculated to satisfy the boundary conditions are very important to determine the entire flow field and the surface forces.

The models proposed in this paper generate a velocity field in good agreement with the Blasius solution for the boundary layer flow over a flat plate, as one can see in the Figs. 6. These figures show converged solution at time t = 2.5 for the *u*-velocity profile as a function of $\eta \equiv y(U/\nu x)^{1/2}$, evaluated at three different *x* stations and compared to the Blasius solution, for Re = 1000, $\Delta t = 0.01$, $\alpha = 0.9$, σ=0.010. The velocity profile present better agreement as *x* approaches the trailing edge of the plate, as expected. Our results also present better agreement to the Blasius solution when compared to the numerical solution obtained by Lewis (1999) for Re = 500, as shown in Fig. 7. The initial convergence evolution of the simulation as time progresses can be seen in Fig. 8.



(a) x = 0.25; (b) x = 0.50; (c) x = 0.75;

Figure 6. Velocity profiles at three *x* stations along the plate; for Re = 1000, $\Delta t = 0.01$, α=0.9, σ=0.010.

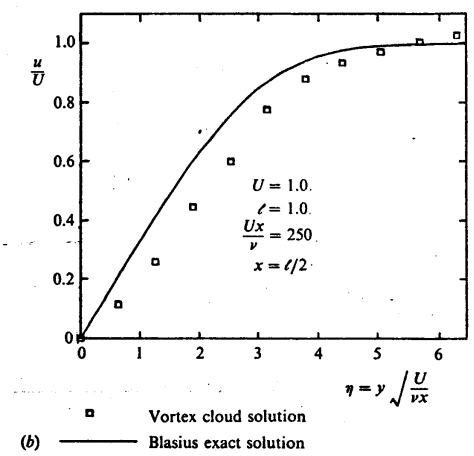


Figure 7. The u-velocity profile at x = 0,5 obtained by Lewis (1991) for Re = 500.

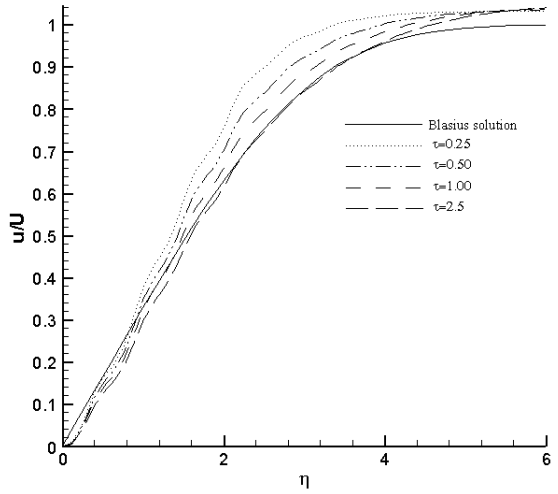


Figure 8. Time convergence for Re =1000.

The wake represented by the cloud of vortices at t = 2.5 is shown in Fig. 9, for Re = 1000. This figure illustrates the growth of the boundary layer along the plate and the high vorticity resolution obtained with our vortex method. Our simulation ends up with N_v = 259,837 vortices at t = 2.5.

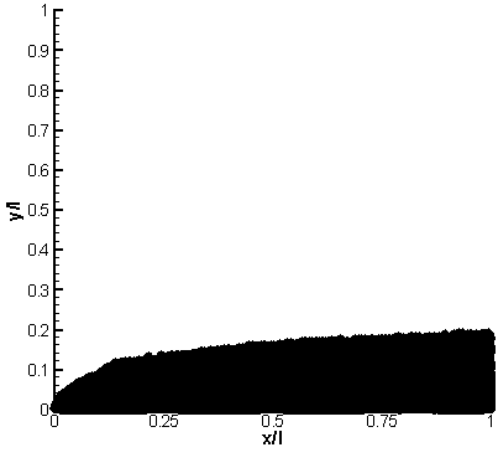


Figure 9. Position of the vortices in the wake at t = 2.5 for Re =1000.

4. CONCLUSIONS

This work introduces several modifications and improvements on previous models for the vortex method to study the complex physical phenomenon associated with convective-diffusive problems such as the boundary layer development over a flat plate. One of the main improvements is the use of the multipole expansion technique through the use of the AFMM, which allows high-resolution simulations to be carried out for longer times. The implementation of this method was necessary in order to replace the stochastic Random Walk Method by the deterministic and more accurate Corrected Core-Spreading Method (Rossi, 1996), which increases the number of vortices in the cloud every time step. The CCSM has proved to give better representation of the vorticity field. Finally, the new vortex generation scheme described in this paper adds an important correction procedure to calculate the vorticity and velocity fields near solid boundaries. As the results for the Blasius flat plate simulation show, the vortex method presented here is very encouraging to describe in detail the high Reynolds-number flows around bodies.

Regardless of the good results shown here, work still needs to be done. The slope of the velocity field near the wall still requires some improvement to be obtained. This information is key to determine the skin friction factor and, consequently, the friction force on the surface of the plate.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge Prof. Leslie Greengard for allowing us to use his original FORTRAN code of the Adaptive Fast Multipole Method, which has been slightly modified by the authors to be used in this work. The authors would also like to acknowledge the Brazilian Army and CNPq, through grants no. 302246/2004-5 and 477950/2004-3, for the financial support of this research project.

6. REFERENCES

Carrier, J., Greengard, L. and Rokhlin, V., 1988, “A Fast Adaptive Multipole Algorithm for Particle Simulation”, SIAM Journal of Scientific Statistics and Computation, Vol. 9, n. 4, pp. 669-686.

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

Greengard, C., 1985, “The core spreading vortex method aproximates the wrong equation”, Journal of Computational Physics, Vol. 61, pp. 345-348.

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

Lewis, R.I., 1991, Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems, Cambridge, Cambridge University Press.

Rossi, L., 1996, “Ressurecting core spreading vortex methods: A new scheme that is both deterministic and convergent”, SIAM Journal, Vol. 17, No. 2, pp. 370-397.

Rossi, L.F., 1997, “Merging computational elements in vortex simulations”, SIAM J. Sci. Comput., vol. 18 , nº 4, pp. 1014-1027.

Santiago, V.S., Bodstein, G.C.R., “Estudo Comparativo de Quatro Diferentes Métodos de Difusão Viscosa para Aplicação no Método de Vórtices”, Paper CIT06-0429, Anais do XI Congresso Brasileiro de Engenharia e Ciências Térmicas - ENCIT (em CD-ROM), Curitiba, PR, ISBN 9788585769, 05 a 08 de dezembro, 2006, pp. 1-12.

Santiago, V. S., Silva, D. F. C., Bodstein, G. C. R., “Análise de Desempenho do Método de Expansão em Multipolos Adaptativo Aplicado a Simulações Numéricas Via Método de Vórtices”, Paper CIT06-0248, Anais do XI Congresso Brasileiro de Engenharia e Ciências Térmicas - ENCIT (em CD-ROM), Curitiba, PR, ISBN 9788585769, 05 a 08 de dezembro, 2006, pp. 1-10.

7. RESPONSIBILITY NOTICE

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