UNIVERSITY OF BRISTOL
DEPARTMENT OF MATHEMATICS

Final Report - SCAT

CLAUDIO ESTEBAN TORRES LÓPEZ

Bristol, United Kingdom
November, 2007

ABSTRACT

This is the final report of the work developed in the Department of Mathematics of the University of Bristol, during a research visit funded under the SCAT project (ref. to `www.scat.bristol.ac.uk`).

The report has four chapters that develop the four major studies made during this project.

The first chapter is about the Birkhoff-Rott equation. This chapter begins with the use of the method of point vortices and goes on to. Moreover include how apply panel method in the equation of Birkhoff-Rott.

The second chapter develops the classic Panel Method and shows a simple example.

The third chapter adds a new approach to solve Boundary Conditions in Vortex Methods, this approach is a conceptual approach and attemp to arrive a consensus in this area.

The last chapter, is a complete study of the spatial adaptation of particles in Vortex Methods and we introduce a new approach to solve the problem in a fast and computational efficient way.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This report develops the work made by the author at the Department of Mathematics of the University of Bristol, United Kingdom, since April-2007 to November-2007.

This report is divided in four chapters, beginning with the Birkhoff-Rott equation and ending with spatial adaptaion in Vortex Methods.

The organization is based on the time line of the study of each topic, i.e. our first work was on the Birkhoff-Rott equation and the last one was spatial adaptation in Vortex Method.

Each chapter was developed independently, but there are some concepts used in a given chapter but developed in another chapter.

This work was supervised by Dr. L. Barba of Department of Mathematics, University of Bristol.

This project was financed by the SCAT project, for more information see `www.scat.bristol.ac.uk`.

Chapter 2

The Birkhoff-Rott Equation

2.1 Abstract

This first chapter deals with the numerical simulation of the Birkhoff-Rott equation, using both point vortices and blobs approximation. It explains what you have to know for a successful numerical simulation of Birkhoff-Rott equation, and it mentions several topics that can improve it, like for example, dynamical insertions, high order blobs, fast multipole method and ODE's solvers.

2.2 Introduction

This chapter presents the Birkhoff-Rott equation, from its beginning to the recent approaches that try to solve it.

We begin with an explanation of what is the Birkhoff-Rott equation, continue with the first known approach by L. Rosenhead `rosenhead1931`, then we mention several other authors that add several improvements, and we develop an idea to apply the panel method to this equation. We finish with the most recent known approach, Krasny's blob method.

2.3 The circular vortex sheet

The circular vortex sheet, from here to forward *CVS*, is modeled by an integrodifferential singular equation that captures the Kelvin-Helmholtz instability, and is known as the Birkhoff-Rott equation.

The equation is:

$$z = z(\gamma, t) \in \mathbb{C} : \quad \frac{\partial \overline{z}}{\partial t} = \frac{1}{2\pi i} \int_0^{2\pi} \frac{d\widehat{\gamma}}{z(\gamma, t) - z(\widehat{\gamma}, t)}, \quad z(\gamma, t_0) = e^{i\gamma}. \tag{2.1}$$

where $i = \sqrt{-1}$ and γ is a Lagrangian parameter. Although we know the analytic solution, which is,

$$z(\gamma, t) = e^{i\left(\gamma + \frac{t}{2}\right)} \tag{2.2}$$

the main reason to study this equation (2.1) is becuase of its chaotic behavior when it is discreticed, and when we perturb its initial condition.

It is important to say that perturbations to the initial condition, like $z(\gamma, 0) = e^{i\left(\gamma + \frac{sin(\gamma)}{2}\right)}$ generate roll-up, and that is fine for a few time-steps of the simulation, but afterward it results in chaos.

The first approach known to solve this problem numerically is the *Point Vortex* approximation; it was introduced by L. Rosenhead in `rosenhead1931`.

The method consists in the discretization of the integral in (2.1), but being careful with singularity by avoiding it.

$$\overline{\left(z\left(\gamma_k, t\right)\right)_t} = \frac{1}{2\pi i} \sum_{\substack{j=0 \\ j \neq k}}^{N} \frac{\Delta\gamma_j}{z\left(\gamma_k, t\right) - z\left(\widehat{\gamma}_j, t\right)} \tag{2.3}$$

In its discretization[1] we lose some part of the integral, for that reason A.I van de Vooren `vandevooren1980` added a term that estimates that area by Taylor expansions. Equation (2.3) becomes,

[1] Note that the *Point Vortices* are the discreticed points of the integral (2.1)

$$\overline{\left(z\left(\gamma_k,t\right)\right)_t} = \frac{1}{2\pi i}\sum_{\substack{j=0\\ j\neq k}}^{N}\frac{\Delta\gamma_j}{z\left(\gamma_k,t\right)-z\left(\widehat{\gamma}_j,t\right)} + \frac{-i\Delta\gamma_k}{4\pi}\frac{\left.\frac{\partial^2 z(\gamma,t)}{\partial\gamma^2}\right|_{\gamma_k}}{\left(\left.\frac{\partial z(\gamma,t)}{\partial\gamma}\right|_{\gamma_k}\right)^2} \tag{2.4}$$

Equation (2.4) contains a full estimation of the hard integral in equation (2.1). It looks well, but the problem now is the calculation of the first and the second derivatives at γ_k.

There are some approaches, like the one of D.W. Moore in `moore1981`. He tries to approximate the derivatives by finite differences, but the chaotic motion of the *Points Vortex*, still appear.

Although later on some new methods appeared, like the Fink-Soh method `finksoh1978` that showed some good result, G. Baker `baker1980` said that the test case used by Fink and Soh was not an adequated one.

Another approach used was the application of a smoothing in data after each time-step. It showed a good behavior and excellent results in the *CVS* with a non-perturbed initial condition, see `longuetETal1976` and `torres2007`, respectively. In the second one, the operator used to smooth the data was built by local polynomials of degree 2 in an overdetermined system solde by least-squares. The Van de Vooren correction was used, as well, but the derivatives were calculated by cubic splines, which provide a better estimation than finite differences.

Another approach(again) uses the well-known *Panel Method*[2]. It was used by Hoeijmakers and Vaatstra `hoeijmakersETal1982`. They show that their method works very well in several problems, and an important concept used is point insertion. But, one disadvantage of the method is that they omit the most difficult part of the vortex sheet, the core of the roll-up, to obtain accurate results. In the same way, Basu et al. `basuETal1995`, did a comparison between *Point Vortices* and the *Panel Method*.

At this point, we want to explain our approach to use the panel method in the Birkhoff-Rott equation (2.1). Let (2.1), and divide it in N sub-domains,

[2]It will be discussed extensively in the next chapters

$$\begin{aligned}&\oint_0^{2\pi} \frac{d\widehat{\gamma}}{z(\gamma,t)-z(\widehat{\gamma},t)} = \\ &\int_{0=\gamma_0}^{\gamma_1} \frac{d\widehat{\gamma}}{z(\gamma,t)-z(\widehat{\gamma},t)} + \int_{\gamma_1}^{\gamma_2} \frac{d\widehat{\gamma}}{z(\gamma,t)-z(\widehat{\gamma},t)} + \cdots + \int_{\gamma_{N-1}}^{\gamma_N=2\pi} \frac{d\widehat{\gamma}}{z(\gamma,t)-z(\widehat{\gamma},t)}\end{aligned} \tag{2.5}$$

Instead of using the trapezoidal rule, we can estimate each integral analytically by the replacement of $z(\gamma,t)$ in each interval by a straight line, like $\alpha_0^{(j)} + \alpha_1^{(j)}\widehat{\gamma}$, where the index j depends on each interval. If we want a higher order approximation, we can replace $z(\gamma,t)$ by a quadratic equation and so on, but for this document a linear approximation is enough. Thus, the analytical integral for a generic interval is,

$$\int_{\gamma_l}^{\gamma_{l+1}} \frac{d\widehat{\gamma}}{z(\gamma,t)-z(\widehat{\gamma},t)} = \frac{\gamma_l-\gamma_{l+1}}{z(\gamma_l,t)-z(\gamma_{l+1},t)} \log\left[\frac{z(\gamma,t)-z(\gamma_l,t)}{z(\gamma,t)-z(\gamma_{l+1},t)}\right] \tag{2.6}$$

where log is the natural logarithm. Therefore,

$$\frac{\partial\overline{z(\gamma,t)}}{\partial t} = \frac{1}{2\pi i} \sum_{\substack{j=0 \\ j\neq k \\ j\neq k-1}}^{N} \frac{\gamma_j-\gamma_{j+1}}{z(\gamma_j,t)-z(\gamma_{j+1},t)} \log\left[\frac{z(\gamma,t)-z(\gamma_j,t)}{z(\gamma,t)-z(\gamma_{j+1},t)}\right] \tag{2.7}$$

Although we still have the problem of the singularity, we can use the Van de Vooren correction again. Even more we can use a higher order correction, using the same idea of Van de Vooren, but we have to be careful because higher order corrections involve derivatives of higher order. It is important to say that all the approaches that I have explained are variations of several mathematical approaches. The next approach is a conceptual approach; although usually criticized by some people that said that that approach is not solving the same problem, there are several researchers that prooved that converge to the original problem. The importance of this method is due to the fact was the first to show roll-up without chaotic result.

The method was proposed by A. Chorin and P. Bernard `chorin1973b`. The main idea can be summarized in that '*when the point character of the point vortices is not taken too literally, the approximation becomes reasonable* '. This means that they approximate the stream function for point vortices that are near, in the Euclidean form.

A. Leonard did an excellent review in `leonard1980`, he shows the big picture of vortex an

particle methods.

Continuing with the idea of Chorin an Bernard, R. Krasny krasny1987 and krasny1986b, introduce the desingularization parameter δ^2.

This applied in the singular term of (2.1) resulting in:

$$\frac{\partial \overline{z\left(\gamma,t\right)}}{\partial t} = \frac{1}{2\pi i}\int_0^{2\pi} \frac{\overline{z(\gamma,t)} - \overline{z(\widehat{\gamma},t)}}{\left|z(\gamma,t) - z(\widehat{\gamma},t)\right|^2 + \delta^2}\, d\widehat{\gamma} \tag{2.8}$$

The singular integral was replaced by a non-singular integral and a parameter δ^2 was added. This smoothing parameter solves the major problem, the singularity. We can discretizy the integral, as follows.

$$\overline{\left(z\left(\gamma_k,t\right)\right)_t} = \frac{1}{2\pi i}\sum_{j=0}^{N} \frac{\overline{z(\gamma_k,t)} - \overline{z(\widehat{\gamma}_j,t)}}{\left|z\left(\gamma_k,t\right) - z\left(\widehat{\gamma}_j,t\right)\right|^2 + \delta^2}\, \Delta\widehat{\gamma}_j \tag{2.9}$$

There are two important things to say about (2.8) and (2.9). The first is that when the parameter $\delta^2 \to 0$ the equation (2.8) converge to (2.1), and in equation (2.9) it is not neccesary to skip the singularity because there is not a division by zero. Hence, the new name of these points is *blobs*. Another important consideration is that the singularity in (2.1), is not only produced when $\gamma \to \widehat{\gamma}$, but when $z(\gamma,t) \to z(\widehat{\gamma},t)$, as well.

Although Krasny's blob method solves an important problem, we have to be careful, because it works well with some considerations.

The main consideration is that for good apprxomation of the real behavior, the distance (in the Euclidean form) in the complex plane between two consecutive blobs must be less than δ, that is the sizes of the blobs. This point is very important, and if it is not followed the simulation does not make any sense. A study of this aspect is found in anderson1985.

Due to the success of the blob method, several authors have shown a big interest, and have published related works, such as J. Ely and G. Baker elyETal1993, S. Kim, J. Lee and S. Sohn in KimETal2003. Its generalization to three dimensional flow by K. Lindsay and R. Krasny lindsay+krasny2001. Other authors modify the kernel (type of blob) to obtain better results, like O. Friendrich and T. Sonar friedrichETal1995, and G. Baker an J. Beale in bakerETal2004 who use exponential kernels.

Also, we would like to mention a comparison of the blob methods that was made by G. Baker and L. Pham in bakerETal2006.

Before concluding we would like to spend some lines on an important point: the insertion of point (vortex or blob) or called remeshing, as well.

To do this task we can use the typical method of interpolation with polynomials, which is how Krasny does it, or splines, or kernels for interpolation like the one known as M_4'that was introduced by J. RMonaghan monaghan1985.

Another important point before concluding is to mention the number of operations for each time step. For all the previous approaches it is $O(N^2)$, where N is the number of blobs or point vortices, and for a large number of blobs or points vortex this is prohibitive. But there is an algorithm that only needs $O(N)$ operations for each time step. It was created by L. Greengard and V. Rokhlin greengard+roklin1987, and is called Fast Multipole Method or FMM. Although we do not explain the algorithm, it is important to mention it.

At this point we can say that the numerical simulation of Birkhoff-Rott's equation is not an easy task. In both case, point vortices and blobs, we have to be careful to obtain successful result.

To finish note that we did not say anything about how to solve the differential part of the equation. We only discussed the integral, that is, the most complicated part of the equation. For the solution of the differential part, the usual method is Runge-Kutta of 4^{th} order, or Euler or Taylor Method.

Chapter 3

Panel Method

3.1 Abstract

The panel method is a well-known technique that allows us solve the problem of bodies inside an incompressible and irrotational fluid. The technique is developed fully and a simple example is shown, we will also show how to increase the order of the method. At the end, some suggestions are added and recent works with the technique are reviewed.

3.2 Introduction

This is a brief description of the Panel Method, not a deep development but enough to understand what it means. The Book *Low Speed Aerodynamics: From Wing. Theory to Panel Methods* by J. Katz and A. Plotkin `katzEtal1991`, will be our main reference.

This chapter develops the idea the panel method beginning with the theoretical concept and where it comes from. Afterwards we show the basic solutions associated to this problem, and the algorithms to use panel method.

A simple example is developed, to explain in a better way the algorithm.

At the end we include a brief list of references to show that the method is still in use.

3.3 Where does it come from?

The main idea of *Panel Methods* is to provide a theoretical development for boundaries inside a fluid, that is a rigid or deformable body in the fluid. Note that the fluid must be incompressible and irrotational, and the continuity equation reduces to (3.1).

$$\nabla^2 \Phi = 0 \tag{3.1}$$

Where Φ is the potential function. The trick here is that the normal component of the velocity to the body's surface must be zero. This means that

$$\nabla \Phi \bullet \overrightarrow{n} = 0 \tag{3.2}$$

Where $\overrightarrow{n}$ is a normal vector to the body's surface, for the no-through condition or $\overrightarrow{s}$ instead of $\overrightarrow{n}$, for no-slip condition, where $\overrightarrow{s}$ is the tangential vector to the body's surface. Note that $\overrightarrow{n}$ is outward.

Thus, we only have to find solutions for (3.1) and because it is a linear equation we only have to superpose them, around the boundaries satisfying (3.2).

3.4 Basic Solutions

3.4.1 Point Source/Sink

The potential in spherical coordinate and located at the origin is (3.3).

$$\Phi(r) = -\frac{\sigma}{4\pi\, r} \tag{3.3}$$

Where σ is the strength of the Point Source, and if $\sigma < 0$ the Point Source becomes in Point Sink.

3.4.2 Point Doublet

The potential in spherical coordinate system is,

$$\Phi(r) = -\frac{\mu}{4\pi}\,\overrightarrow{n} \bullet \nabla\left(\frac{1}{r}\right) \tag{3.4}$$

It can be interpreted as the limit of going together a point source and point sink. And the μ is the strength of the Point Doublet.

3.4.3 Polynomials

The polynomials are the easiest solution of the Laplace equation, for instance,

$$\Phi(x, y, z) = A\,x + B\,y + C\,z \tag{3.5}$$

and polynomials of higher degree with some considerations,

$$\Phi(x, y, z) = A\,x^2 + B\,y^2 + C\,z^2 \tag{3.6}$$

for instance, we can replace (3.6) in (3.1), and obtain (3.7).

$$\nabla^2\Phi = A + B + C \tag{3.7}$$

Let $B = 0$ in (3.7) and obtain (3.8).

$$\begin{aligned} A + C &= 0 \\ A &= -C \end{aligned} \tag{3.8}$$

then we obtain (3.9).

$$\Phi(x, y, z) = A\left(x^2 - z^2\right) \tag{3.9}$$

3.4.4 Point Vortex

For this one the coordinate system will be the Cartesian one, and the potential for 2D is:

$$\Phi(x,y) = \frac{-\Gamma}{2\pi} ArcTan\left(\frac{y-y_0}{x-x_0}\right) \tag{3.10}$$

where Γ is the strength of the point Vortex, and x_0 and y_0 are the center of the Point Vortex.

3.4.5 An extra consideration

It is important mention that each one of the previous *Basic Solution* has a stream function ψ associated. And can be easily obtained by the next equations,

$$\begin{aligned} u &= \frac{\partial\Phi}{\partial x} = \frac{\partial\Psi}{\partial y} \\ v &= \frac{\partial\Phi}{\partial y} = -\frac{\partial\Psi}{\partial x} \end{aligned} \tag{3.11}$$

Where u is the x component of the velocity, v is the y component of the velocity, Φ is the potential function and Ψ is the stream function.

Another important thing is that when we plot Φ or Ψ for constants values, we will get the lines of constant potential and streamlines, respectively.

3.5 The Algorithm

- Selection of singularity element, this selection means that we have to select one solution from the basic solutions. Note that we can mix different basic solutions.
- Discretization of geometry (and grid generation), this means generate a grid around the body or bodies that are in the fluid. Later on, we have to select the *collocation points*, where zero normal velocity of flow will be enforced, and it should be on the surface of the body.
- Influence coefficients, in this phase the coefficients are calculated at the collocation points. The unknowns will be the strength of each element of singularity located around the body, it will be our matrix A of our future system ($Ax = b$) to solve.
- Establish RHS (or b), it will be the influence of the free stream associate to the fluid product dot normal vector at the collocation point.

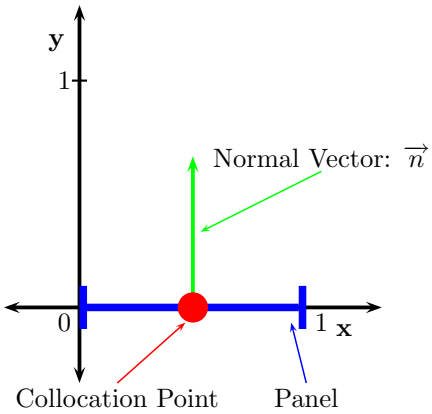

Figure 3.1: Figure of a panel, collocation point and normal vector

- Solve the linear system of equations, for this step we suggest use *gmres* with a preconditioner. But, we suggest be cautious due that the system is ill conditioned.

3.6 A Simple Example: The flow around a cylinder in 2D

- *Selection of singularity element*: Point Vortex
- *Discretization of geometry*: straight around the cylinder
- *Influence coefficient*: This is the point that usually take more time but is not the more hard

For this simple example we only will show how calculate the influence coefficients from a panel to itself and to the others.

Let have a panel centered at origin (for simplicity) and with unitary length, see figure (3.1). As we have said, we have to eliminate the normal component of the velocity at collocation point. For our example the collocation point was located at the center of the panel (but can be located where do you want), excluding the end, because this is the usual place for the

singularities.

But, before to continue we have to convert our potential function to velocity field, with the equation (3.11), that is:

$$\begin{aligned} \frac{\partial \Phi}{\partial x} &= u = \frac{\Gamma}{2\pi} \frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2} \\ \frac{\partial \Phi}{\partial y} &= v = -\frac{\Gamma}{2\pi} \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} \end{aligned} \tag{3.12}$$

And for our example $x_0 = 1$ and $y_0 = 0$, and our vector $\overrightarrow{n} = (0, 1)$. Thus, we can easily calculate the influence of this point vortex to own collocation point and to the other collocations points.

This is the procedure to calculate the influence coefficients for panel method of zero order. But, there are two ways to increase the order, and we will mention them briefly.

The first one is generate a continuous distribution of point vortices on the panel, and this means that we have to calculate (3.12) again, and it will be

$$\begin{aligned} u &= \frac{\Gamma}{2\pi} \int_0^1 \frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2} dx \\ v &= -\frac{\Gamma}{2\pi} \int_0^1 \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} dx \end{aligned} \tag{3.13}$$

And the second one is build by linear distribution of Γ (the strength) above the panel, like $\Gamma(x) = \Gamma_0 + \Gamma_1 \, x$.

- *Establish RHS*: this point is easy see from the equation (3.2), where Φ is total potential from the singularity elements and the freestream, this means that the equation (3.2) becames (3.14).

$$\nabla \left(\Phi_{singularity} + \Phi_{freestream} \right) \bullet \overrightarrow{n} = 0 \tag{3.14}$$

And de RHS (Right hand side) is,

$$RHS = -\nabla \Phi_{freestream} \bullet \overrightarrow{n} \tag{3.15}$$

In other words, the free stream velocity at collocation point dot the normal vector of that collocation point for all collocation points.

- *Solve the linear set of equation*: this is the hardest point because the system is ill conditioned, and when we increase the order of the panel method the conditions number become a little bit better.

Afterward the usually secondary computation comes, like pressures, loads, velocity, and so on.

Before conclude it is important to mention that the origin of panel method was for boundary conditions, but there are anothers applications like in vortex sheet by Hoeijmakers, HWM. and Vaatstra, W. in hoeijmakersETal1982, even more, there are recent approach that improve it, like as ramachandranETal2002 by Ramachandran, P. and Rajan, S. C. and Ramakrishna, M., that does faster the two-dimensional panel method, or C. Lee and J.Kerwin in LeeETal2003 and J. Kouh and J. Suen in kouhETal2001 that try a desingularization for more accuracy, among others.

Chapter 4

Boundary Conditions in Vortex Methods

4.1 Abstract

Boundary in Vortex Method is not a new problem or forgotten problem, but doesn't have a consensus in how solve it. For this reason there are several approach or technique that try solve it. Our try to arrive to concensus with Koumoutsakos, Leonard and Pépin's approach, but trying alleviate the hard task of calculate a Vortex Sheet, even more how transfer the strength of it to the blobs.

4.2 Introduction

The boundary conditions are a very interesting topic in Vortex Method because either of the actual methods have a total concensus in the area, but there are several successful approach, since A. J. Chorin in chorin1973, P. Koumoutsakos, A. Leonard, and F. Pépin in koumoutsakos+leonard+pepin1994,L. Rossi in rossi1995, among others. But, is important say that all of them are based in the Lighthill's model lighthill1963.
The main idea of this chapter is provide another conceptual approach for boundaries, this is

mainly based in the book cottet+koumoutsakos2000 by G. Cottet and P. Koumoutsakos, and from here on will be called *the Classic Method*.

We encourage read the book cottet+koumoutsakos2000 to understand in a better way the problem.

The chapter is developed from the explanation of the classic method to the develop of the our conceptual approach, for both of them there are several figures to a better understand.

4.3 The Classic Method

We have to say that it will be a conceptual revision of the classic method.

The classic method in 5 step, from the *Initial Configuration* to *New state, satisfying boundaries*.

- Initial configuration, see figure (4.1)
- **[1st step]** Convection(Biot-Savart law), see figure (4.2)
- **[2nd step]** Diffusion(core spread), see figure (4.3)
- **[3rd step]** Generate *Vortex Sheet* at (*Boundary*), see figure (4.4)
- **[4th step]** Transfer *Strength* from *Vortex Sheet* to *blobs* near boundary, see figure (4.5)
- **[5th step]** Remeshing[1], if is necessary, see figure (4.6)
- New state, satisfying boundaries

4.4 Our New Conceptual Approach

We agree with the classic method, but our suggest or conceptual approach try answer the question *Why does teh Vortex Sheet necessary?*, and our answer is that that Vortex Sheet is not necessary and the only necessary is the '$\Delta\gamma$'from the $4th$ step of classic method, showed in figure (4.5).

[1] It will be developed on the next chapter

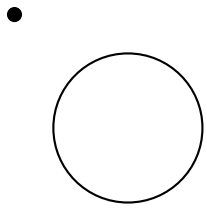

Figure 4.1: Initial Configuration

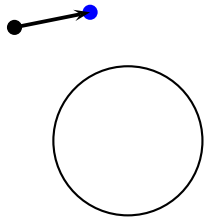

Figure 4.2: **[1st step]** Convection

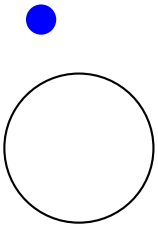

Figure 4.3: **[2nd step]** Diffusion

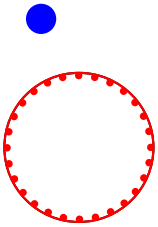

Figure 4.4: **[3rd step]** Generate *Vortex Sheet* at Boundary

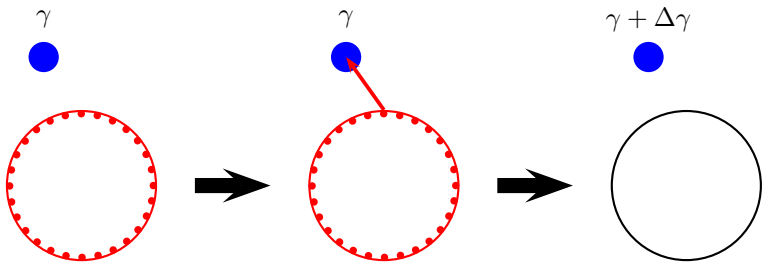

Figure 4.5: **[4th step]** Transfer *Strength* from *Vortex Sheet* to *Blobs* near boundary

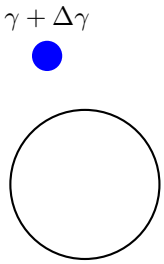

Figure 4.6: **[5th step]** Remeshing (if is necessary)

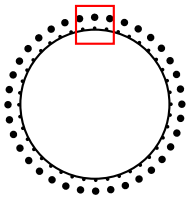

Figure 4.7: **[1st step, our approach]** Select blobs near boundary and collocations points. The red square is the place where we will show a zoom for the next step.

4.5 How can we do it?

We can do it by the same concept used in how calculate the Vortex Sheet (by Panel Method), we have to keep the collocation points but the unknowns will be the '$\Delta\gamma$'directly, instead of the strength of Vortex Sheet. With this new approach we don't have to fight with how transfer the strength to blobs near the boundary or how calculate the Vortex Sheet, note that we don't say that new system that we have to solve be easy, that is other problem.

4.6 The New Conceptual Approach on a Cartoon

Before continue, we have to say that the next three step replace the $4th$ and $5th$ step in the classic method.

- **[1st step]** Select blobs near boundary and collocation points, see figure (4.7)
- **[2nd step]** Choose between no-slip or no-through condition, and generate the new system, see figure (4.8)
- **[3rd step]** Solve the system

We know that boundaries in Vortex Method is a hard task, we believe that our conceptual approach is a drop in the sea of knowledge.

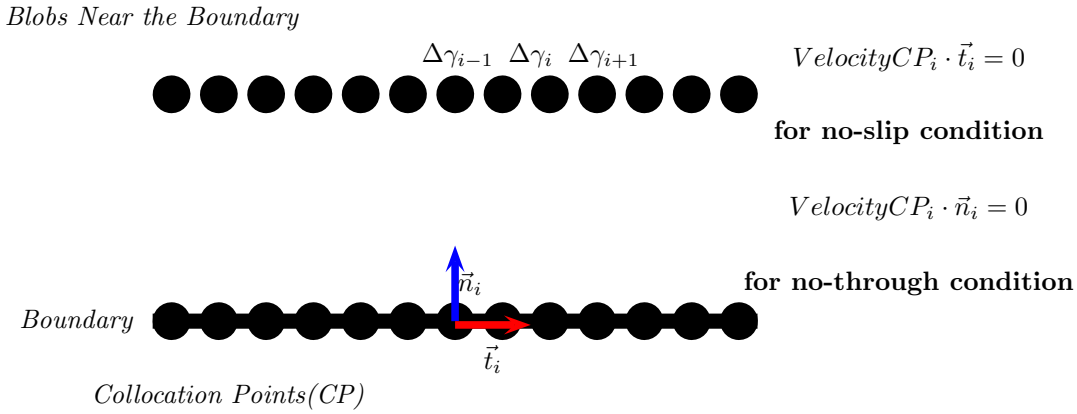

Figure 4.8: **[2nd step, our approach]** Choose between no-slip or no-through condition, and generate the new system. Note that it is a zoom from the red box in Figure 4.7

Chapter 5

An Iterative Spatial Adaptation Method

5.1 Abstract

The spatial adaptation in vortex methods, i.e. an accurate spatial adaptation is a bottle neck. The main reason of it is because the method used to solve that is very expensive in a computational way, although there are other that are very fast but are poor accurate, i.e. the error associated is not forgettable. We present a new approach to obtain an accurate spatial adaptation in a very fast way.

5.2 Introduction

The success of vortex method depends directly of what supposition carry out. One of them is the overlap between the particles, which is changing with the time. And here is the importance of spatial adaptation, which solves the problem.

Before continue, we encourage read an excellent PhD Thesis where the concept used here were totally explained, it was written by Dr. L. Barba [barba2004].

This chapter begins with the description of the problem, afterward the explanation of what

is a particle?[1], the explanation of the two options used to solve the problem. At the end, we will show our method and some result of the application of it.

5.3 The Problem

In vortex method we have a vorticity field that we want describe as a sum of particles placed around the domain with define overlap, in other words, we want the circulation of each particle located over the domain, satisfying the overlap to describe the vorticity field.

5.4 What is a particle?

A particle in simply words is a radio basis function and has a size. For instance, a Gaussian particle is (5.1) and the plot of it, see figure (5.1), where $k = 2$ and $\sigma = 1$.

$$\mathbb{K}_\sigma(r) = \frac{1}{k\pi\sigma^2} \exp\left(\frac{-|r|^2}{k\sigma^2}\right) \tag{5.1}$$

5.5 How Do Several Particles Represent the Vorticity Field?

The particles represent the vorticity field by the sum of them, situated over the domain.
Let the vorticity field showed in figure (5.2), and for instance we can represent it with three particles, see figure (5.3), and analytically it can be expressed by (5.2), where $V(x)$ is the vorticity field represented by the three particles, and γ_0, γ_1 and γ_2 are the circulation ones associated to each particle.

$$V(x) = \gamma_0 \mathbb{K}_\sigma(x - x_0) + \gamma_1 \mathbb{K}_\sigma(x - x_1) + \gamma_2 \mathbb{K}_\sigma(x - x_2) \tag{5.2}$$

[1] It is a element used later on

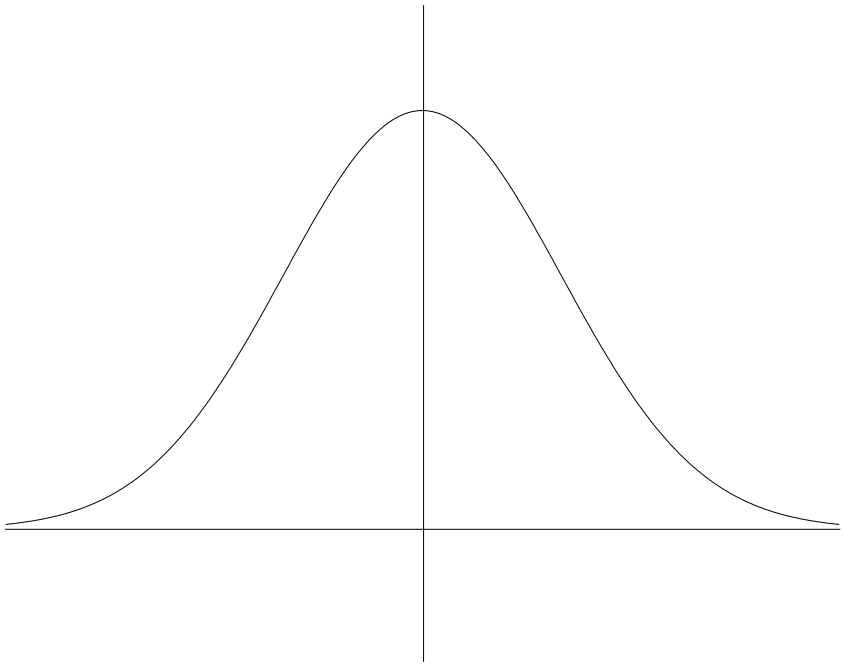

Figure 5.1: A Gaussian particle

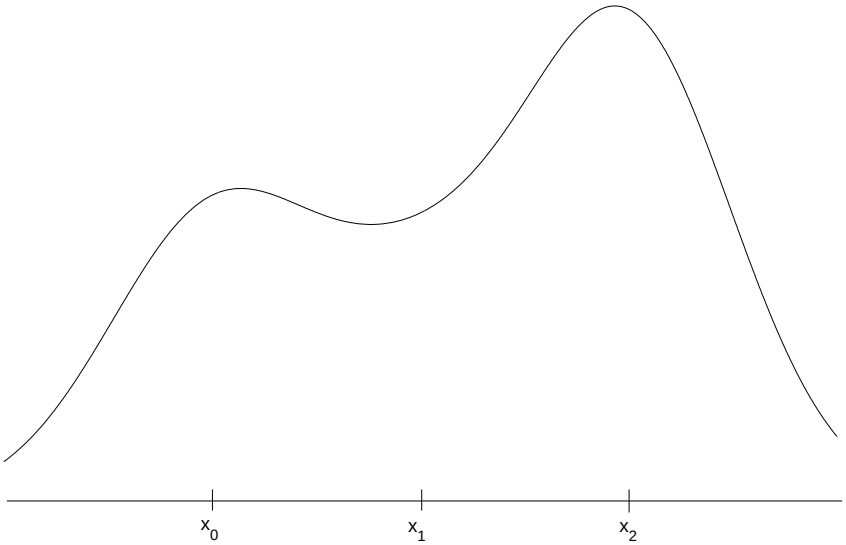

Figure 5.2: Vorticity Field

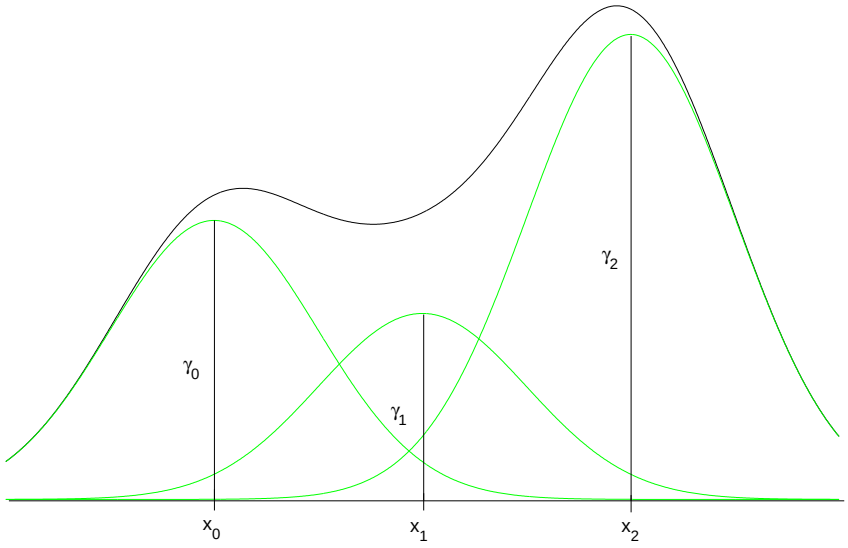

Figure 5.3: Vorticity field with particles

5.6 How Can We Obtain the Circulation of Each Particle?

We have two options, **Fast-Cheapest but not good** or **Slow-Expensive but very good**. Before continue, we have to say that the size(σ) for all particles will be the same. It is not a strong constrain, it is only for simplicity and can be generalized easily.

Before continue we have to define h, which is:

$$h = overlap \cdot \sigma \tag{5.3}$$

5.6.1 Fast-Cheapest but not good

This way to estimate the circulation is by the equation (5.4), where j is the particle's index. For the figure (5.3) and (5.2) the value of j are $0, 1$ and 2.

$$\gamma_j = V(x_j) \cdot h^2 \tag{5.4}$$

This is really fast, even more the order of the algorithm for N particles is $O(N)$ and the

memory usage[2] is $O(N)$, but the relative L2 error[3] is approximate 10^{-3}, and the error is the main disadvantage of this method.

5.6.2 Slow-Expensive but very good

This option consist in solve the equation (5.2) like as an interpolation problem, it means build the system $Ax = b$, where the right hand side (RHS) b is the vorticity field evaluated at the centre of each particle, the unknown x is the circulation and A is the matrix of influence.
The problem with this option is that we want solve by SVD or QR or LU or any of that sort of factorization, the order of the algorithm are $O(N^3)$ where N is the number of particles. Others sort of algorithms to solve the this problem are the iterative ones, like **gmres** and so on, where these type of algorithms have order $O(N^2\, M)$, where M is the number of iterations. Either of the previous algorithm use an order $O(N^2)$ of memory, and it is a **BIG** bottle neck, although we can write this approach in a free matrix way and solve the problem of the memory usage we still have the a quadratic or cubic order in the algorithm.
And finally the last problem is the conditioning of the matrix A, that is very ill-conditioning. A usually good idea is use a preconditioner, but Which one?!.
The *good* thing with it, is that the final L2 error that we will obtain is order $10 - 14$ being careful with the conditioning.

5.7 Our Approach

The idea of our method is solve the equation (5.2), i.e. obtain very good result but with and iterative method that only need $O(N)$ in memory usage and $O(N^2\, M)$ in computational cost, where N is the number of particles and M is the number of iterations. We have to add that we know by computational simulations that $M << N$ and with the fast Gauss transform (FGT) (`greengardETal1991`), it can be order $O(N)$!.

[2]Saving the array of vorticity values in memory
[3]It is important say that there are other methods in the same way that are a little bit better

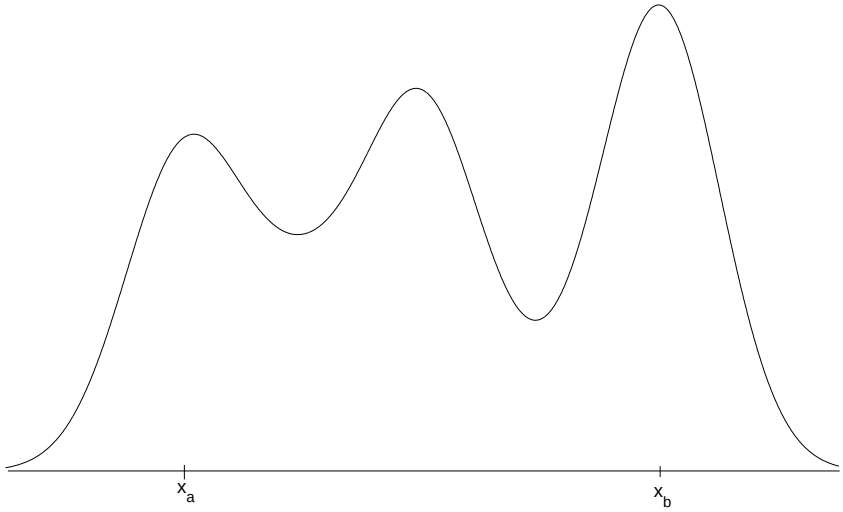

Figure 5.4: Vorticity field for example

5.8 How?

By the divide-and-conquer strategy and for an easy understand we will show an example.

Let $V(x)$ the vorticity field (5.4) in the domain $x_a \leq x \leq x_b$ where we want to reproduce the same vorticity field (or almost) with five particles.

Thus, we can place the particles over the domain like (5.5) to satisfy constrain max. distance between particles h.

Thus, with (5.4) we can estimate the circulation of each particle, see figure (5.6).

And we know that a fast estimation of vorticity field has a relative L2 error of order 10^{-3}, but we can improve it iteratively, where this estimation will be our first guess. The idea to improve the solution is solve the system related with (5.2) but keeping only some unknowns and the others convert in constrain, in other word, replace some unknowns for the initial guess and keep the others ones as a unknowns. For our example (5.5) means convert it in (5.6), where c_1, c_2, c_3, c_4 and c_5 are the column of the matrix A is the matrix of influence defined in (5.5), and $\widehat{\gamma_0}$ and $\widehat{\gamma_4}$ are unknowns replaced by the initial guess calculated by (5.4).

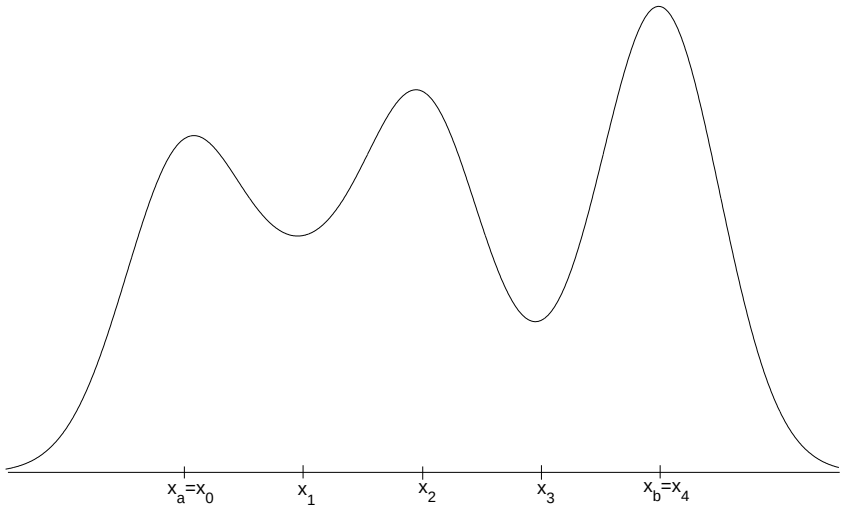

Figure 5.5: Vorticity field satisfying constrain max. distance between particles h for Example

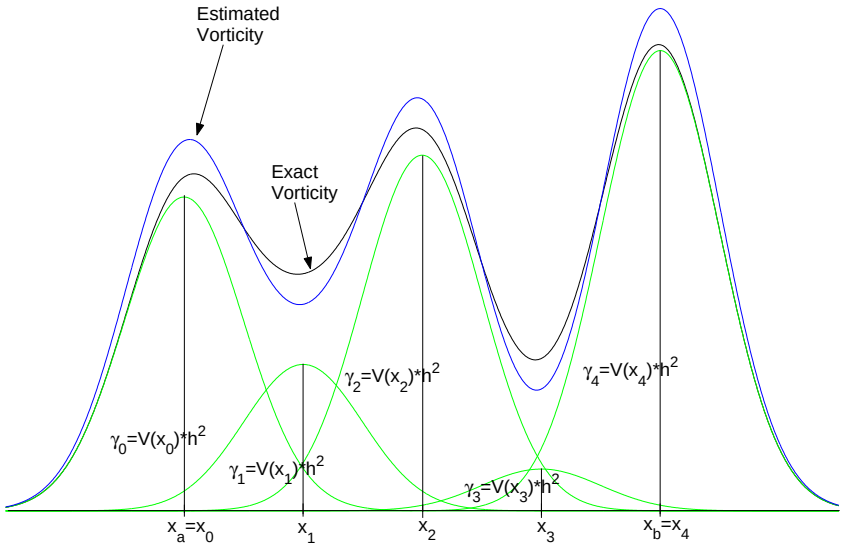

Figure 5.6: Vorticity field satisfying constrain max. distance between particles h for Example

$$\underbrace{\begin{pmatrix} \mathbb{K}_\sigma(x_0 - x_0) & \mathbb{K}_\sigma(x_0 - x_1) & \dots & \mathbb{K}_\sigma(x_0 - x_4) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{K}_\sigma(x_4 - x_0) & \mathbb{K}_\sigma(x_4 - x_1) & \dots & \mathbb{K}_\sigma(x_4 - x_4) \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} V(x_0) \\ V(x_1) \\ V(x_2) \\ V(x_3) \\ V(x_4) \end{pmatrix}}_{b} \tag{5.5}$$

$$\left(\begin{array}{c|c|c|c|c} c_1 & c_2 & c_3 & c_4 & c_5 \end{array}\right) \begin{pmatrix} \widetilde{\gamma_0} \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \widetilde{\gamma_4} \end{pmatrix} = \begin{pmatrix} V(x_0) \\ V(x_1) \\ V(x_2) \\ V(x_3) \\ V(x_4) \end{pmatrix} \tag{5.6}$$

We know that can not keep any constant in our vector x of unknowns, for this reason we must move it to the RHS, thus our system became in (5.7),

$$\underbrace{\left(\begin{array}{c|c|c} c_2 & c_3 & c_4 \end{array}\right)}_{\widehat{A}} \underbrace{\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix}}_{\widehat{x}} = \underbrace{\begin{pmatrix} V(x_0) \\ V(x_1) \\ V(x_2) \\ V(x_3) \\ V(x_4) \end{pmatrix} - \left(\begin{array}{c|c} c_1 & c_5 \end{array}\right) \begin{pmatrix} \widetilde{\gamma_0} \\ \widetilde{\gamma_4} \end{pmatrix}}_{\widehat{b}} \tag{5.7}$$

Although is not a bad idea try solve (5.7), but we have to add an simplification, it mean keep only the rows in (5.7) where our unknown (γ_1, γ_2 and γ_3) have the centre of its particle. But before continue we have to rewrite (5.7) in (5.8) for an easier understand,

$$\left(\begin{array}{c} r_1 \\ \hline r_2 \\ \hline r_3 \\ \hline r_4 \\ \hline r_5 \end{array}\right) \left(\begin{array}{c} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array}\right) = \widehat{b} \tag{5.8}$$

thus is easier see which row eliminate, for our example means eliminate r_1 and r_5, and the result is,

$$\underbrace{\left(\begin{array}{c} r_2 \\ \hline r_3 \\ \hline r_4 \end{array}\right)}_{\widehat{\widehat{A}}} \underbrace{\left(\begin{array}{c} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array}\right)}_{\widehat{\widehat{x}}} = \underbrace{\left(\begin{array}{c} V(x_1) \\ V(x_2) \\ V(x_3) \end{array}\right) - \left(\begin{array}{c} r'_2 \\ \hline r'_3 \\ \hline r'_4 \end{array}\right) \left(\begin{array}{c} \widehat{\gamma_0} \\ \widehat{\gamma_1} \end{array}\right)}_{\widehat{\widehat{b}}} \tag{5.9}$$

and now we have to solve our small system (5.9) instead of the big one. But that is not all, when we have the value of γ_1, γ_2 and γ_3 we have drop-off γ_1 and γ_3, and only save the value of γ_2. It new value of γ_2 will be our initial guess for the next iteration of our iterative algorithm. But before we have to solve another four small system to obtain the new values of γ_0, γ_1, γ_3 and γ_4, by the same way.

Before continue, we have to mention that the Fast Gauss Transform can be used in the calculus of $\widehat{\widehat{b}}$ in (5.9), where we have a sum of exponential[4].

[4]For this case where the particles are exponential

5.9 An Explanation/Justification of Each Step

The first one, from (5.5) to (5.8) is mainly because we want solves the system (5.5) by the resolution of small system, and this step allow that.
The second one, from (5.8) to (5.9) is only by cost-profit reason, based in our numerical simulations. We solved the big least-square system (5.8) and doesn't give us enough profit compared with its cost, in other word, we are fitting the exponential only in the place where we keep the unknowns.
And the third one, where we only save the value of γ_2 of the solution of (5.9), is based in the idea that the unknowns dropped are near to the unknowns converted in constrain and has a big influence of them but this influence is less as far we are of them and this means that the unknowns far of them are significant better.
And, of course, we have to iterate the algorithm to converge, but the convergence is very fast and the results are excellent.

5.10 The Algorithm

Before describe the algorithm we have to define some parameters to generalize the algorithm.

- **Block**: This will be the block in the domain where we will improve the solution and we will save the values.
- **Number of particles per block side**: This will be the min number of particles allowed per side of block, it can be bigger but not lower.
- **Relative Buffer Size**: This will be the relative size respect to the size of the block to include particles as buffer unknowns. This means if we used 1 we will include 9 blocks surrounding the **Block**, as we show in the figure (5.8).

The algorithm based in the previous example and the previous definitions is:

1. Select the variables (γ_1, γ_2 and γ_3) that will be the unknowns (This means select **Number of particles per block side** and **Relative Buffer Size**).

2. Obtain an estimation of the value of the variables that will become in constrain ($\widehat{\gamma_0}$ and $\widehat{\gamma_4}$) and we will call **Support Unknowns** (It is estimate a value of the circulation of the particles that are not in the **Block** or associated to **Relative Buffer Size**).

3. Build the small system $\widehat{\widehat{A}}\widehat{\widehat{x}} = \widehat{\widehat{b}}$ explained in (5.9), where the unknowns are in the **Block** and associated to **Relative Buffer Size**.

4. Save the value of the **Selected Unknown** (γ_2) and dropp-off the value of the **Buffer Unknowns** (γ_1 and γ_3) (Save only the values of the unknowns in the **Block**).

5. Come back to step 1 and select another set of unknowns, to obtain and update of the all unknowns.

6. And now with a better estimation of the unknowns, we can iterate the algorithm from 1 to 6 to improve the solution again.

7. Finish the algorithm when arrive in a define error or the algorithm converge.

5.11 Set Up of Numerical Simulations

The case of study is based in the well-known *Lamb-Oseen Vortex*[5] given by the equation (5.10) where $\omega(r,t)$ is the vorticity field, r is the radio from where it is centered, ν is the viscosity, Γ_0 is an own parameter and t is the time.

$$\omega(r,t) = \frac{\Gamma_0}{4\,\pi\,\nu\,t}\,\exp\left(-\frac{r^2}{4\,\pi\,t}\right) \tag{5.10}$$

The Lamb-Oseen located at the center of the complex plane where r becomes $|z|$, i.e. the norm of our complex variable, and the domain is $[-1,1]\times[-1,1]$ for the real and imaginary part, respectively.

The parameters of (5.10) for our numerical simulations can be seen in table (5.1).

And the number of particles used is 15876, with the parameters showed in the table ((5.2), distributed uniformly over the domain.

[5]Do not mix up this with the particles used, although they look similar are two things totally different.

Γ_0	=	1
ν	=	0.1
t	=	1

Table 5.1: Parameters for the Lamb-Oseen for our Case of Study

$overlap$	=	1
σ	=	0.1
Number of particles per block side	=	5
Relative Buffer Size	=	1.6

Table 5.2: Parameters of particles

We have to mention that **Block** is dependent of the **Number of particles per block side** and is a concept.

The domain was distributed uniformly, see the figure (5.7).

Thus we can select the *Selected Unknowns (Block)*, *Buffer Unknowns (particles associated to Relative Buffer size)* and *Support Unknowns (the remain particles)* very easy, see figure (5.8).

5.12 The Numerical Simulations

5.12.1 Evolution of Error

For each iteration we illustrate two $\log_{10}$ plot graphs of absolute of relative error of the vorticity, the left illustration is with a fixed scale to compare the error with the others iterations and the right illustration is with a variable scale to see the details in each iterations. See the figures from (5.9) to (5.15).

5.12.2 Perturbation Initial Guess

Another numerical simulation that we did, was add a small perturbation to the initial guess. The purpose of this numerical simulation is to watch the dependency of the algorithm to its initial guess.

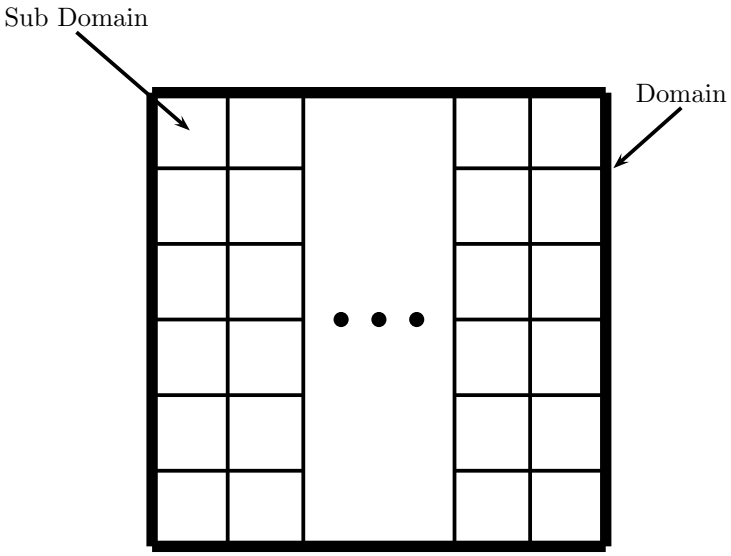

Figure 5.7: Domain - SubDomain

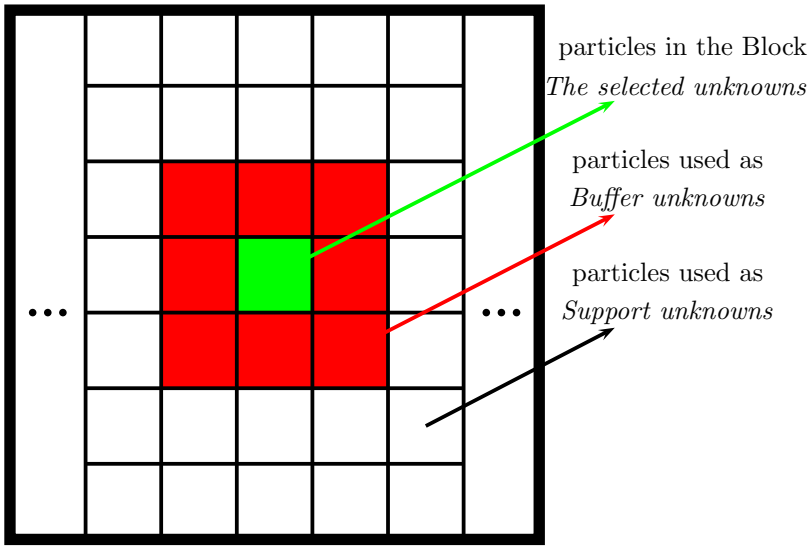

Figure 5.8: Location of Select, Buffer and Support Unknowns on the Domain

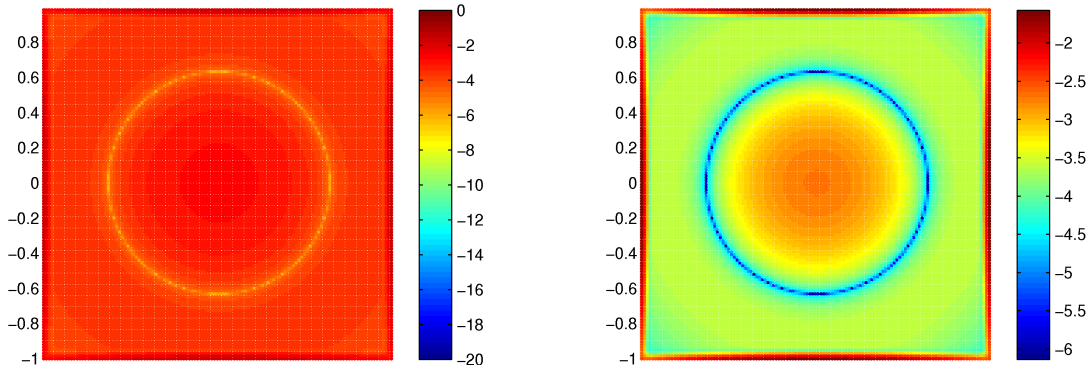

Figure 5.9: **Iteration 0** First estimation of circulation with (5.4) and the Log_{10} |relative L2 error of vorticity| $= -2.1064$

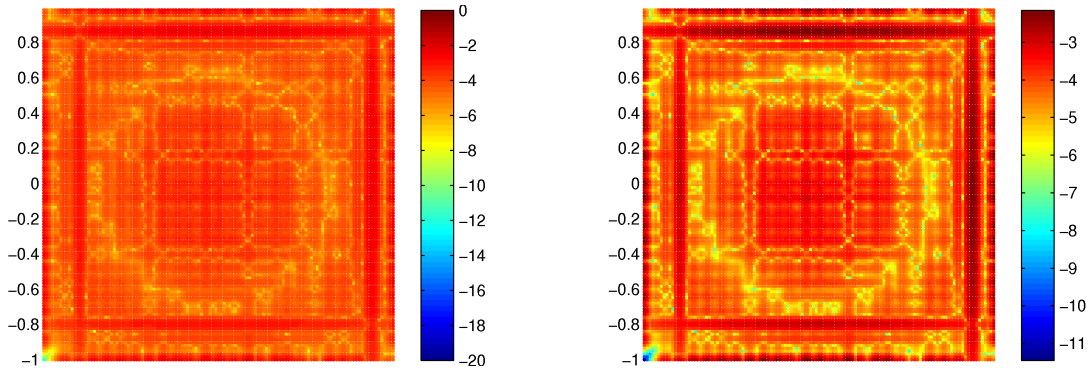

Figure 5.10: **Iteration 1** Log_{10} |relative L2 error of vorticity| = −2.6876

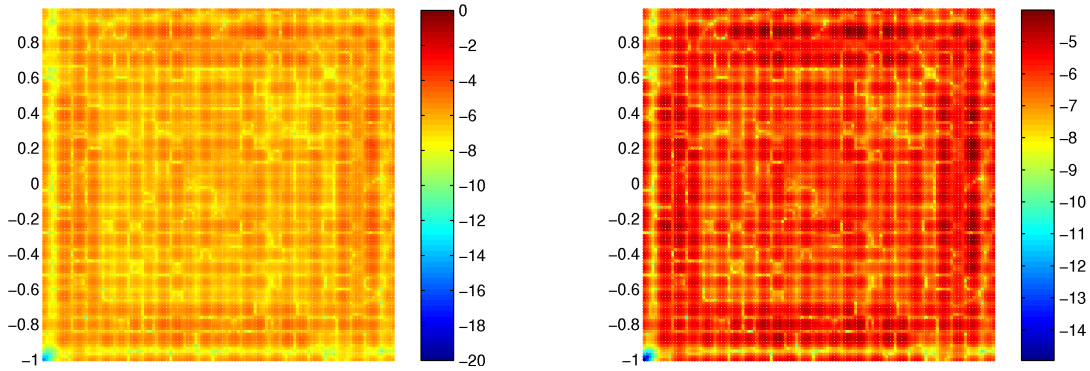

Figure 5.11: **Iteration 4** Log_{10} |relative L2 error of vorticity| = −4.7806

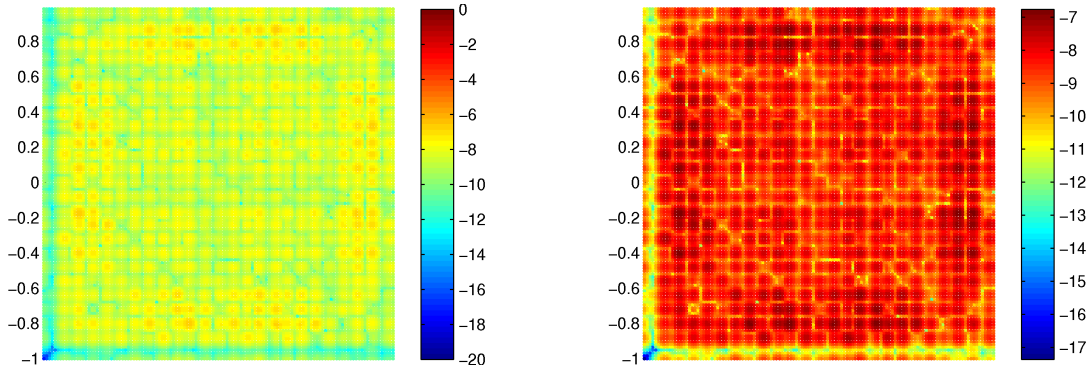

Figure 5.12: **Iteration 8** Log_{10} |relative L2 error of vorticity| = −7.3515

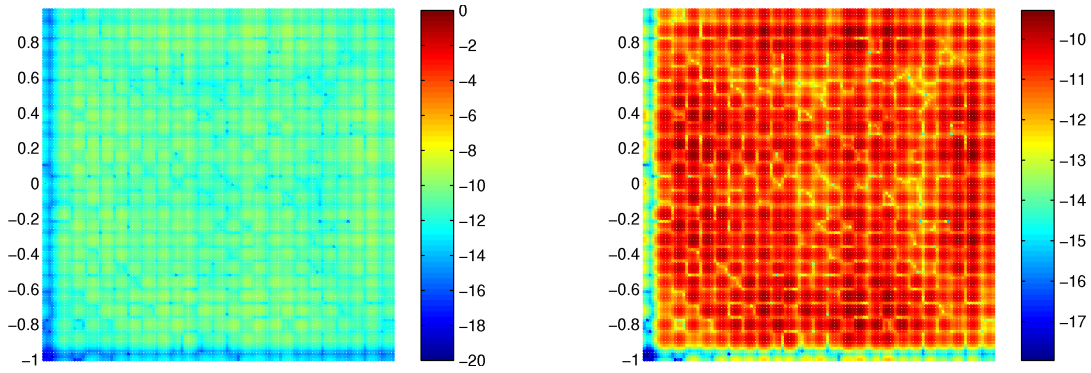

Figure 5.13: **Iteration 12** Log_{10} |relative L2 error of vorticity| = −9.8305

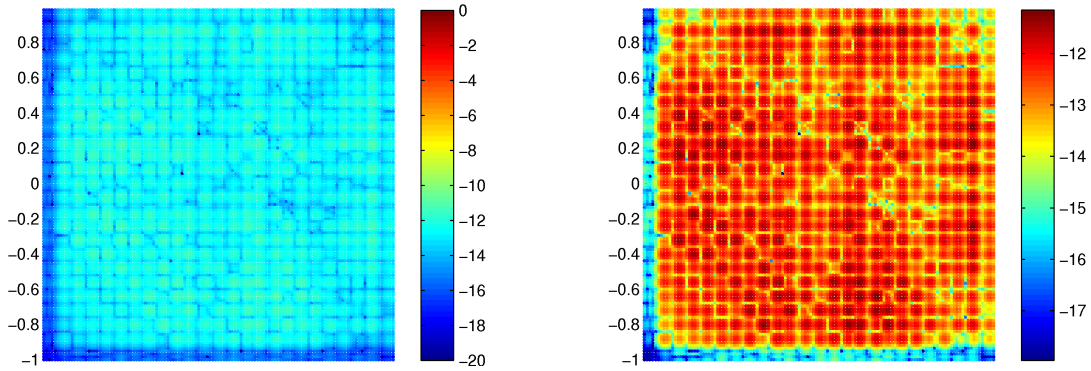

Figure 5.14: **Iteration 15** Log_{10} |relative L2 error of vorticity| = −11.6355

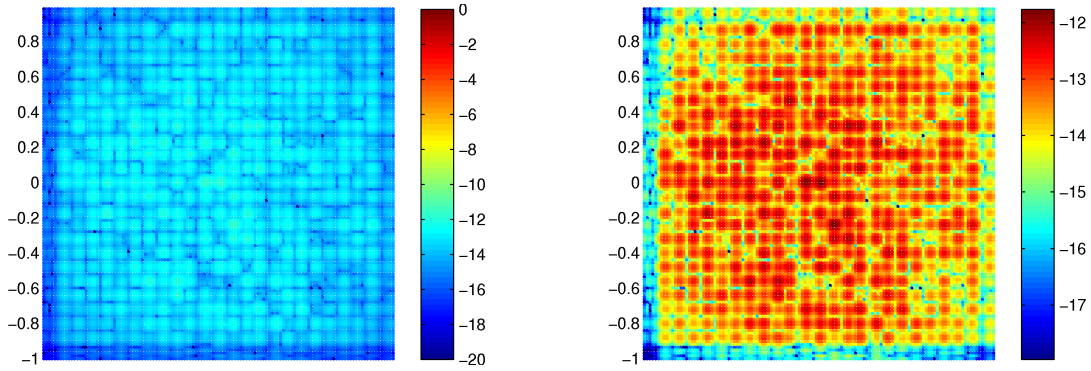

Figure 5.15: **Iteration 17** Log_{10} |relative L2 error of vorticity| = −12.4561

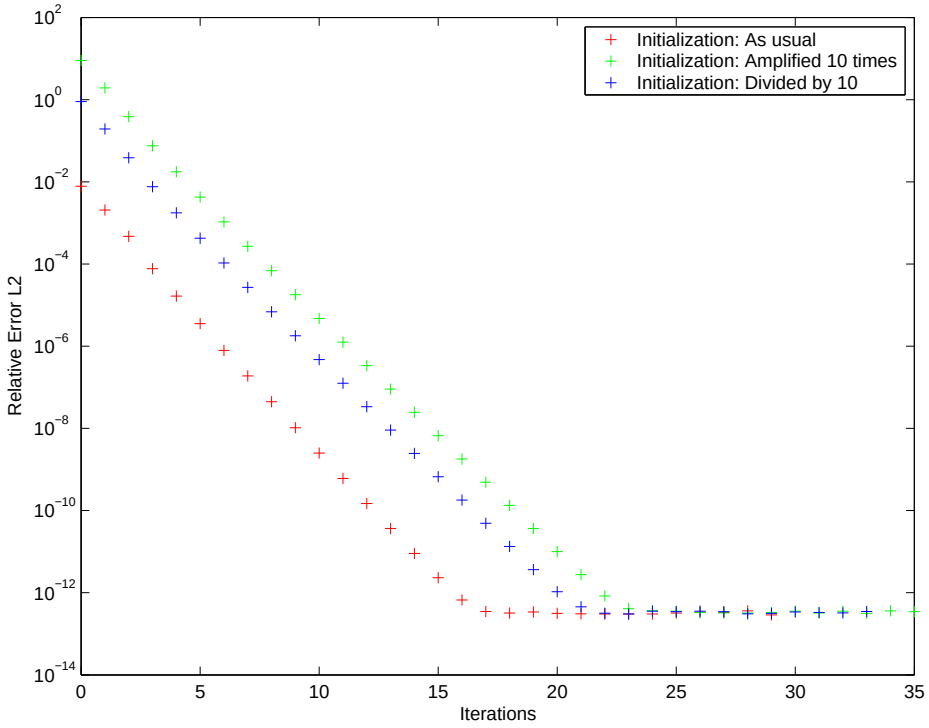

Figure 5.16: Perturbation Initial Guess

For first one the initial guess was amplified 10 times. For the second one the initial guess was divided by 10.

In the figure (5.16), we can see the relative L2 error of vorticity. And for the both of them the initial guess got worse, that is normal, and the estimation without perturbation is approximated 10^{-3}, how we said.

The important thing, is that the algorithm converge at the same error taking more iterations for both of them, and that is acceptable, i.e if it begin with worse initial guess, it will take more iterations to converge.

Note that the number of iterations is very low compared to the number of total particles and the scale used to plot the relative L2 error is logarithmic.

Chapter 6

Conclusions, Recommendations and Further Work

The results obtained are very good, but we need more time to understand it.

Although we developed four chapters, we will center the conclusions in the last one, due that we obtained the best result. The other ones need more work or have been developed a lot by several authoresses, and our study was essential enlightening. However, we believe that the third chapter has a good potential, as the last chapter.

We are very sure that our result in the last chapter was not a particular one. We believe that is the begin of something better.

And we can conclude that the strategy of divide and conquer has a good application in this sort of problem, even more the successful here can be extrapolated in other fields. But, we have to add that this approach is not a typical strategy divide and conquer, it has some special considerations like the buffer of unknowns and we do not forget the global dependence.

Thus, we can say that using our developed and being carefully with the parameters we can obtain excellent results of spatial adaptation in vortex methods. Even more, we show that a perturbation in the initial guess does not destroy or considerable affect the algorithm, it showed that the algorithm has a stable behavior.

Our recommendations are that our method should be used in this sort of problem, and is

very dependent of the parameters used. If you want use this method in other field or context, it can generate bad result or not. We know that a hard modification of the parameters can create a divergence algorithm, but with the right parameters it converge very fast. Note that if you follow our develop of the algorithm you can and omit the last steps, you can build a general algorithm that does not use the feature of the particles decay far away.

The further work is understand the importance of the buffer of unknowns and how parameters affect the convergence of the algorithm and how it works with another sort of vorticity. We know that the Lamb-Ossen is a very particular case and is time to fight with other sort of vorticity field.

References

anderson1985 .....

C. Anderson.

A vortex method for flows with slight density variations.

J. Comp. Phys., 61:417–444, 1985.

baker1980 .....

G. Baker.

A test of the method of fink and soh for following vortex-sheet motion.

J. Fluid Mech., 100:209–220, 1980.

bakerETal2004 .....

G. R. Baker and J. T. Beale.

Vortex blob methods applied to interfacial motion.

J. Comp. Phys., 196:233–258, 2004.

bakerETal2006 .....

G. R. Baker and L. D. Pham.

A comparison of blob methods for vortex sheet roll-up.

J. Fluid Mech., 547:297–316, 2006.

barba2004 .....

L. A. Barba.

Vortex method for computing high-Reynolds number flows: Increased accuracy with a fully mesh-less formulation.

PhD thesis, California Institute of Technology, 2004.

basuETal1995 .....

A. J. Basu, R. Narasimha, and A. Prabhu.

Modelling plane mixing layers using vortex points and sheet.

Applied Mathematical Modelling, 19:66–75, 1995.

chorin1973b .....

A. J. Chorin.

Discretization of a vortex sheet, with an example of roll-up.

J. Comp. Phys., 13:423–429, 1973.

chorin1973

A. J. Chorin.

Numerical study of slightly viscous flow.

J. Fluid Mech., 57:785–796, 1973.

cottet+koumoutsakos2000

G.-H. Cottet and P. Koumoutsakos.

Vortex Methods. Theory and Practice.

Cambridge University Press, 2000.

elyETal1993

J. S. Ely and G. R. Baker.

High-precision calculations of vortex sheet motion.

J. Comp. Phys., 111:275–281, 1993.

finksoh1978

P. T. Fink and Soh W. K.

A new approach to roll-up calculations of vortex sheets.

Proc. R. Soc. Lond. A, 362:195–209, 1978.

friedrichETal1995

O. Friedrich and T. Sonar.

Regularizing discretizations of the birkhoff-rott equation.

Mathematical methods in the applied science, 18:615–625, 1995.

greengard+roklin1987

L. Greengard and V. Rokhlin.

A fast algorithm for particle simulations.

J. Comp. Phys., 73:325–348, 1987.

greengardETal1991

L. Greengard and J. Strain.

The fast gauss transform.

SIAM J. Sci. and Stat. Comp., 12:79–94, 1991.

hoeijmakersETal1982

HWM. Hoeijmakers and W. Vaatstra.

A higher order panel method to vortex sheet roll-up.

AIAA J., 21(4):516–524, 1982.

katzEtal1991 .....

J. Katz and A. Plotkin, editors.

Low-Speed Aerodynamics: From Wing Theory to Panel Methods.

McGraw-Hill Companies, 1991.

KimETal2003 .....

S. Kim, J. Lee, and S. Sohn.

A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow.

J. Phy. Soc. of Japan, 72:1968–1976, 2003.

kouhETal2001 .....

J. Kouh and J. Suen.

A 3d potential-based and desingularized high order panel mathod.

Ocean Engineering, 28:1499–1516, 2001.

koumoutsakos+leonard+pepin1994 .....

P. Koumoutsakos, A. Leonard, and F. Pépin.

Boundary conditions for viscous vortex methods.

J. Comp. Phys., 113:52–61, 1994.

krasny1986b .....

R. Krasny.

Desingularization of periodic vortex sheet roll-up.

J. Comp. Phys., 65:292–313, 1986.

krasny1987 .....

R. Krasny.

Computation of vortex sheet roll-up in the Trefftz plane.

J. Fluid Mech., 184:123–155, 1987.

LeeETal2003 .....

C. Lee and J. Kerwin.

A B-spline higher-order panel method applied to two-dimensional lifting problem.

J. of Ship Research, 47:290–298, 2003.

leonard1980 .....

A. Leonard.

Vortex methods for flow simulation.

J. Comp. Phys., 37:289–335, 1980.

lighthill1963 .....

M. J. Lighthill.

Introduction: Boundary layer theory.

In *Laminar Boundary Layers*, pages 46–113. Oxford University Press, 1963.

lindsay+krasny2001 .....

K. Lindsay and R. Krasny.

A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow.

J. Comp. Phys., 172:879–907, 2001.

longuetETal1976 .....

M. S. Longuet-Higgins and E. D. Cokelet.

The deformation of steep surface waves on water . 1 . a numerical method of computation.

Proc. R. Soc. Lond. A, 350:1–26, 1976.

monaghan1985 .....

J. J. Monaghan.

Extrapolating B-splines for interpolation.

J. Comp. Phys., 60:253–262, 1985.

moore1981 .....

D. W. Moore.

On the point vortex method.

SIAM J. Sci. Stat. Comput, 2(1):65–84, 1981.

ramachandranETal2002 .....

P. Ramachandran, S. C. Rajan, and M. Ramakrishna.

A fast, two-dimensional panel method.

SIAM J. Sci. Comput., 24:1864–1878, 2002.

rosenhead1931 .....

L. Rosenhead.

The formation of vortices from a surface of discontinuity.

Proc. R. Soc. Lond. A, A134:170–192, 1931.

rossi1995 .....

L. Rossi.

Vortex computations of wall jet flows.

In *Proceedings of the Forum on Vortex Methods for Engineering Applications*. Sandia National Labs., 1995.

Albuquerque, NM.

torres2007 .....

C. E. Torres.

Computación en cluster: estabilización y desacoplamiento computacional del problema de la capa límite círcular.

Master thesis, Universidad Técnica Federico Santa María, Valparaíso, Chile, 2007.

vandevooren1980 .....

A. I. van de Vooren.

A numerical investigation of the rolling-up of vortex sheet.

Proc. R. Soc. Lond. A, A373:67–91, 1980.