

MASTER OF SCIENCE THESIS

Hybrid Eulerian-Lagrangian Vortex Particle Method

A fast and accurate numerical method for 2D Vertical-Axis
Wind Turbine

L. Manickathan B.Sc.

Date TBD

Faculty of Aerospace Engineering · Delft University of Technology

Hybrid Eulerian-Lagrangian Vortex Particle Method

**A fast and accurate numerical method for 2D Vertical-Axis
Wind Turbine**

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace
Engineering at Delft University of Technology

L. Manickathan B.Sc.

Date TBD



Copyright © L. Manickathan B.Sc.
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
AERODYNAMICS AND WIND ENERGY

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **“Hybrid Eulerian-Lagrangian Vortex Particle Method”** by **L. Manickathan B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: Date TBD

Head of department:

prof.dr.ir. G.J.W. van Bussel

Academic Supervisor:

dr.ir. C.J. Simao Ferreira

Academic Supervisor:

dr.ir. A. Palha da Silva Clerigo

Industrial Supervisor:

prof.dr.ir. I. Bennett

Summary

This is the summary of the thesis.

Acknowledgements

I wish to thank the following persons...

Delft, The Netherlands
Date TBD

L. Manickathan B.Sc.

Contents

Summary	v
Acknowledgements	vii
List of Figures	xiv
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
1.1 Motivation and Goal	1
1.2 Research Aim and Plan	2
1.3 Introduction to Hybrid Eulerian-Lagrangian Vortex Particle Method . . .	2
1.3.1 Advantage of domain decomposition	2
1.3.2 Methodology	2
1.4 Thesis Outline	2
2 Lagrangian Domain: Vortex Particle Method	3
2.1 Introduction to Vortex Particle Method	3
2.1.1 Vorticity	3
2.1.2 Velocity-vorticity formulation of the Navier-Stokes equations . . .	4
2.1.3 Viscous splitting algorithm	4
2.2 Spatial Discretization: Generation of Vortex Blobs	5
2.2.1 Biot-Savart law	5
2.2.2 Discrete form of vorticity field	6
2.2.3 Convection of vortex blobs	6
2.2.4 Mollified vortex kernels	6
2.2.5 Vortex blob initialization	8

2.2.6	Remeshing scheme: Treating lagrangian grid distortion	11
2.3	Diffusion of Vortex Methods	13
2.3.1	Modified remeshing for treating diffusion	14
2.4	Boundary conditions at solid boundary	17
2.4.1	Boundary integral equations	18
2.4.2	Panel method for treating no-slip boundary condition	20
2.4.3	Convergence study of panel method	23
2.5	Simulation acceleration techniques	23
2.5.1	Fast multi-pole Method	23
2.5.2	Parallel computation in GPU	23
2.6	Validation of lagrangian method	23
2.6.1	Lamb-oseen vortex at $Re = 100$	23
2.6.2	Convergence study of the viscous vortex method	23
2.6.3	Convection of Clercx-Bruneau dipole at $Re = 625$	23
2.7	Summary	23
3	Eulerian Domain: Finite Element Method	25
3.1	Introduction to Finite Element Method	26
3.1.1	Finite element discretization	26
3.1.2	Finite element function and function space	26
3.2	Solving the Finite Element problem	26
3.2.1	Introduction to FEniCS Project	26
3.2.2	Mesh generation using GMSH	26
3.3	Solving Incompressible Navier-Stokes Equations	26
3.3.1	Velocity-pressure formulation	26
3.3.2	Incremental pressure correction scheme	26
3.3.3	Determining the vorticity field	26
3.3.4	Determining the body forces	26
3.4	Validation of eulerian method	26
3.4.1	Clercx-Bruneau dipole collison at $Re = 625$	26
3.4.2	Impulsively started cylinder at $Re = 550$	26
3.5	Summary	26
4	Hybrid Eulerian-Lagrangian Vortex Particle Method	27
4.1	Theory of Domain Decomposition Method	27
4.1.1	Advantage of domain decomposition	27
4.1.2	Assumptions and Limitations	27
4.1.3	Modified coupling strategy	27
4.2	Eulerian-Lagrangian coupling algorithm	27
4.2.1	Eulerian dirichlet boundary condition	27
4.2.2	Vorticity interpolation algorithm	27
4.3	Introduction to pHyFlow: Hybrid solver	27
4.3.1	Program structure	27
4.4	Summary	27

5	Verification and Validation of Hybrid Method	29
5.1	Error in coupling: Verification with Lamb-Ossen vortex	30
5.1.1	Generation of artificial vorticity	30
5.2	Clercx-Bruneau dipole convection at $Re = 625$	30
5.2.1	Comparison of vorticity contours	30
5.2.2	Variation in maximum vorticity	30
5.2.3	Variation in kinetic energy	30
5.2.4	Variation in enstrophy	30
5.3	Clercx-Bruneau dipole collision at $Re = 625$	30
5.3.1	Comparison of vorticity contours	30
5.3.2	Variation in maximum vorticity	30
5.3.3	Variation in kinetic energy	30
5.3.4	Variation in enstrophy	30
5.3.5	Variation in palinstrophy	30
5.4	Impulsively started cylinder problem at $Re = 550$	30
5.4.1	Evolution of the wake	30
5.4.2	Evolution of pressure and friction drag	30
5.4.3	Evolution of lift	30
5.5	Moving body	30
5.5.1	Error due to perturbation lag	30
5.6	Proof of concepts	30
5.6.1	Multiple cylinder case	30
5.6.2	Stalled airfoil at $Re = 5000$	30
5.7	Summary	30
6	Conclusion and Recommendation	31
6.1	Conclusion	31
6.1.1	Lagrangian domain	31
6.1.2	Eulerian domain	31
6.1.3	Hybrid method	31
6.2	Recommendations	31
6.2.1	Lagrangian domain	31
6.2.2	Eulerian domain	31
6.2.3	Hybrid method	31
	References	33

List of Figures

2.1	Circulation of the fluid	4
2.2	Vortex blob with Gaussian distribution: $[k = 2, \sigma = 1.0]$	7
2.3	Vortex blob with overlap σ/h	8
2.4	Mollified vorticity field of an arbitrary vorticity function with overlap = 1.0, $\sigma = 0.19$, $h = 0.19$. Vortex blob strength has been assigned by equation 2.20, sampling at exact vorticity [\bullet , red dot]. Figure depicts exact vorticity distribution ω [—, solid], vorticity field of each blob ω_i [—, green dashed], the mollified vorticity field ω^h [- -, dashed].	9
2.5	Mollified vorticity field after two Beale's iteration, overlap = 1.0, $\sigma = 0.19$, $h = 0.19$. Figure depicts exact vorticity distribution ω [—, solid], vorticity field of each blob ω_i [—, green dashed], the mollified vorticity field ω^h [- -, dashed].	10
2.6	Convergence of vorticity by modifying the spatial resolution. Figure depicts exact vorticity field ω with [—, black] and various resolutions.	10
2.7	Lagrangian distortion of the vortex blobs after 100 steps. The initial vorticity field $\omega(\mathbf{x}, 0) = \exp(-12 \mathbf{x})$ with $\Delta t = 0.1$, $\sigma = 0.02$ and overlap = 1.0. Figure depicts (a) the initial and (b) the final distribution of the vortex blobs.	11
2.8	Remehsing of vortex blob (\bullet , green) on the uniform grid defined by the (4×4) 2-D stencil.	12
2.9	Monaghan's [13] third-order peicewise smooth B-spline kernel, M'_4	13
2.10	One dimensional, Simple redistribution scheme, diffusing the vortex blobs to the four stencil points ($k = i - 1, \dots, i + 2$) with vortex blob positioned at $x_i \leq x_\nu \leq x_{i+1}$ and nominal blob spacing h	16
2.11	Extended vorticity field consisting of vorticity in the fluid and vortex sheet distribution around the body.	18
2.12	Extended vorticity field: Vortex sheet being an extension to the vorticity field (resolved by the vortex blobs) which is able to fully resolve the boundary vorticity	19
2.13	Vortex panel's global (a) and local (b) coordinates system definition as defined by Katz and Plotkins [9].	22

2.14 twoPanelBodies 22

List of Tables

Nomenclature

Latin Symbols

c^2	Diffusion parameter	$[-]$
\mathcal{E}	Enstrophy	$[m^2 \cdot s^{-2}]$
h	Nominal particle spacing	$[m]$
h_ν	Characteristic diffusion distance	$[m]$
\mathbf{K}	Biot-Savart kernel	$[-]$
\mathbf{K}_σ	Vortex blob kernel	$[-]$
\mathbf{A}	Vortex panel influence matrix	$[-]$
k_d	Diffusion frequency multiple	$[-]$
$\hat{\mathbf{n}}$	Normal vector	$[-]$
N_p	Number of particles	$[-]$
overlap	Overlap ratio of the blobs	$[-]$
p	Pressure	$[\text{Pa}]$
$\hat{\mathbf{s}}$	Tangent vector	$[-]$
t	Time	$[s]$
\mathbf{u}	Velocity	$[m \cdot s^{-1}]$
\mathbf{u}_b	Velocity of the body	$[m \cdot s^{-1}]$
\mathbf{u}_γ	Vortex sheet induced velocity	$[m \cdot s^{-1}]$
\mathbf{u}_{ext}	External induced velocity	$[m \cdot s^{-1}]$
\mathbf{u}^h	Discrete velocity	$[m \cdot s^{-1}]$
\mathbf{u}_∞	Free-stream velocity	$[m \cdot s^{-1}]$
\mathbf{u}_ϕ	Potential velocity	$[m \cdot s^{-1}]$

\mathbf{u}_{slip}	Boundary slip velocity	$[m \cdot s^{-1}]$
\mathbf{u}_{ω}	Vortical velocity	$[m \cdot s^{-1}]$
W	Interpolation kernel weight	$[-]$
\mathbf{x}	Position vector	$[m]$
\mathbf{x}_{ν}	Position vector of particle to be diffused	$[m]$
\mathbf{x}_p	Position vector of the particle	$[m]$

Greek Symbols

α_p	Circulation of the particle	$[m^2 \cdot s^{-1}]$
β_p	Corrected circulation of the particle	$[m^2 \cdot s^{-1}]$
Γ	Circulation	$[m^2 \cdot s^{-1}]$
Γ_b	Circulation of moving body	$[m^2 \cdot s^{-1}]$
γ	Vortex sheet strengths	$[s]$
Γ_{γ}	Circulation of vortex sheet	$[m^2 \cdot s^{-1}]$
Γ_{ω}	Circulation of the fluid	$[m^2 \cdot s^{-1}]$
Δt_c	Convection time-step size	$[s]$
Δt_d	Diffusion time-step size	$[s]$
ϵ	Distance between the particles	$[m]$
ζ_{σ}	Smooth cut-off function of the blob	$[-]$
ν	Kinematic viscosity	$[m^2 \cdot s^{-1}]$
ξ	Scale relative position of particle to stencil node	$[-]$
ρ	Density	$[kg \cdot m^{-3}]$
σ	Core size	$[m]$
ω	Vorticity	$[s^{-1}]$
ω^h	Discrete vorticity field	$[s^{-1}]$

Abbreviations

CSVM	Constant-Strength Vortex Method
FMM	Fast-Multipole Method
GPU	Graphics Processing Units
LSTSQ	Least-Square solution method
PSE	Particle Strength Exchange
VAWT	Vertical-Axis Wind Turbine
VPM	Vortex Particle Method
VRM	Vortex Redistribution Method

Chapter 1

Introduction

Conventional energy resources such as fossil fuels and nuclear energy are not only limited supply but also pose adverse effects on the environment. Therefore, we are striving to find a cheap and renewable source of energy. Wind energy is such source of energy and is therefore getting more popular and also become more affordable and novel renewable technologies such as Vertical-Axis Wind Turbine (VAWT) is now an interested research field.

Vertical-Axis Wind turbines are unlike the normal wind turbine. Typical wind turbines are mounted on a mast away from the ground and generates energy by spinning normal to the ground. However, a VAWT spins parallel to the ground with its hub located at the ground [21]. The advantages of the vertical axis wind turbine are what makes them ideal for a source of renewable energy. As the turbine is located at the ground (unlike the Horizontal-Axis Wind Turbine), it is easily accessible and can be easily maintained. The second main advantage of the VAWT is the way it dissipates its wake [8] [19]. As the fluid past the turbine is more turbulent, the flow is able to smooth out much earlier. This means that it possible to places VAWTs much closer to each other is so in future this means that a VAWT farm can potentially give more power per area. Furthermore, operate independent of the flow direction and can operate at low wind speeds (low tip-speed ratios).

1.1 Motivation and Goal

However, with these advantages also comes drawbacks. As the blades passes through its own dirty air (the wake), complex wake-body interactions take places. These have adverse effect on the blade structure and therefore is more susceptible to fatigue. This happens because the blades are constantly pitching in front the free-stream flow and complex flow behaviour such as dynamic stall and constant vortex shedding occurs [16]. This complex fluid behaviours makes it hard to predict the performance of a VAWT and this is one of the reasons why VAWTs are not mainstream. In addition, as the VAWT operates at

large Reynolds number, accurate numerical methods are computationally very expensive. Therefore, it is vital to have a good understanding of the flow structure evolution and the wake generation of the VAWT using not only an efficient method, but also an accurate one.

To summarize, we are now able to formulate a research goal. The key interest of this project is to develop an efficient, reliable, and an accurate numerical method for modelling the flow around a 2D VAWT. For now, only 2D problems are considered because 3D method is build upon the methodology of the 2D. Thus, once the 2D methodology is made, a 3D numerical method should be a straightforward extension.

Furthermore, the numerical method efficient at capturing both the near-wake phenomenons such as the vortex shedding, dynamic stall, & the wake-body interaction, and should be able to capture the large scale flow structure such as the evolution of the VAWT wake. From this criterias, we are able to formulate the research question.

1.2 Research Aim and Plan

Research Question: *Is it possible to develop a numerical method that is both efficient at capturing the small-scale phenomenons and the large scale phenomenons? Is it possible to apply this to a 2D VAWT?*

Research aim and plan:

- Develop a numerical method for capturing small-scale phenomenons and large scale phenomenons.
- Ensure this tool is efficient, reliable, and accurate.
- Verify, Validate the tools with model problems.
- Apply the model to the 2D flow of VAWT.

With the above formulate research question, aim and plan we are able to thoroughly perform the literature study to determine whether the research goal stated here is feasible. Finally, this report will answer why a Hybrid Eulerian-Lagrangian Vortex Particle Method will be used to the achieved the goals.

1.3 Introduction to Hybrid Eulerian-Lagrangian Vortex Particle Method

1.3.1 Advantage of domain decomposition

1.3.2 Methodology

1.4 Thesis Outline

Lagrangian Domain: Vortex Particle Method

2.1 Introduction to Vortex Particle Method

Vortex Particle Method ([VPM](#)) is a branch of computational fluid dynamics that deals with the evolution of the vorticity of the fluid in a lagrangian description. Typically, the fluid is viewed at a fixed window where it is described as a function of space \mathbf{x} and time t . However, the lagrangian point of view regards the fluid as a collection of the particles carrying the property of the fluid.

Unlike the typical eulerian method that require discretization of all the fluid domain, VPM only needs fluid elements where there is vorticity. This means that the VPM are inherently auto-adaptive method that only simulated the flow of interest. Furthermore, with the computational acceleration methods such as Fast-Multipole Method ([FMM](#)) and parallel computation on Graphics Processing Units ([GPU](#)) , VPM can be more efficient than typical eulerian methods.

2.1.1 Vorticity

Vorticity ω , the governing element of vortex particle method, is defined as

$$\omega = \Delta \times \mathbf{u}, \quad (2.1)$$

where \mathbf{u} is the velocity. The circulation Γ is defined as

$$\Gamma = \int_L \mathbf{u} \cdot d\mathbf{r} = \int_S \omega \cdot \mathbf{n} dS, \quad (2.2)$$

by the stokes theorem, as represents the integral vorticity of the domain, figure [2.1](#)

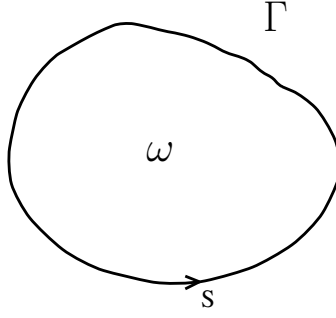


Figure 2.1: Circulation of the fluid

2.1.2 Velocity-vorticity formulation of the Navier-Stokes equations

The governing equation of the vortex particle method is velocity-vorticity $\mathbf{u} - \omega$ formulation of the Navier-Stokes equations [5]. The 2-D incompressible Navier-Stokes momentum equation is given as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad (2.3)$$

relating the velocity field $\mathbf{u}(\mathbf{x}, t)$ to the pressure field $\mathbf{p}(\mathbf{x}, t)$, the kinematic viscosity ν and density ρ . Furthermore, we also have to satisfy the incompressibility constraint given as

$$\nabla \cdot \mathbf{u} = 0. \quad (2.4)$$

To attain the velocity-vorticity formulation, we should take the curl of the velocity-pressure $\mathbf{u} - p$ formulation of the Navier-Stokes equation. Taking the curl of the momentum equation 2.3, we get the vorticity transport equation

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \nabla^2 \omega, \quad (2.5)$$

which only relates the vorticity to the velocity enabling us to neglect the pressure field. Note that as we are dealing with the two dimensional flow, we neglected the stretching term.

2.1.3 Viscous splitting algorithm

Vortex particle method was initially used to model the evolution of incompressible, inviscid flows. However, in order to simulate a real flow, we must also deal with the viscous behaviour of the fluid. Chorin [3] has shown that using the viscous splitting algorithm, it is possible to simulate a viscous flow.

The viscous splitting algorithm is basically a fractional step method, where the viscous and the inviscid part of the transport equation is dealt in two subsequent steps,

- Sub-step 1: convection

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = 0; \quad (2.6)$$

- Sub-step 2: diffusion

$$\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega. \quad (2.7)$$

The first sub-step of the evolution deals with the convection of the vorticity. Note that, by convection we imply the advection of the vorticity field where the diffusion process is neglected. The second sub-step is where we deal with the diffusion of the vorticity field.

There are several advantages to this type of evolution. As the convection and diffusion are handled separately, there is minimum dispersion during the convection and furthermore, there is no restriction of the advection CFL number [20].

There are many ways of dealing with the diffusion of the vorticity field. During this project, we use a modified interpolation kernel [20] that can simultaneously treat diffusion and remesh the vortex particles, see section 2.3.

2.2 Spatial Discretization: Generation of Vortex Blobs

In order to deal with the vorticity field, we must first discretize the vorticity to vortex particles. Vortex blobs have been first introduced by Chorin and is a mollified particle carrying the local circulation. Vortex blobs describe a smooth vorticity field and are ideal because of it does not cause singularity issues when particles approach each other.

!!! check for consistency, continuity !!!

2.2.1 Biot-Savart law

The velocity field can be decomposed using the Helmholtz decomposition, given as

$$\mathbf{u} = \mathbf{u}_\omega + \mathbf{u}_\phi, \quad (2.8)$$

where \mathbf{u}_ω is the rotational component of the velocity and \mathbf{u}_ϕ is the irrotational component, solenoidal and potential velocity respectively. In an unbounded flow we have \mathbf{u}_ϕ equal to the free-stream velocity \mathbf{u}_∞ . For bounded flow, we must include the presence of the body, see section 2.4.

The velocity can be related to the vorticity using the Biot-Savart law

$$\mathbf{u}_\omega = \mathbf{K} \star \omega, \quad (2.9)$$

where the \star represents convolution of the 2-D kernel \mathbf{K} given by

$$\mathbf{K} = \frac{1}{2\pi |\mathbf{x}|^2} (-x_2, x_1). \quad (2.10)$$

2.2.2 Discrete form of vorticity field

The spatial discretization of the fluid domain is done through N quadrature points. With the Biot-Savart law, we can treat these quadratures as discrete particles carrying the local quantities. The discrete vorticity field is given as

$$\omega(\mathbf{x}, t) \simeq \omega^h(\mathbf{x}, t) = \sum_p \alpha_p(t) \delta[\mathbf{x} - \mathbf{x}_p(t)], \quad (2.11)$$

where α_p is the estimate of the circulation around the particle \mathbf{x}_p with core size σ . We must not that ω^h is an approximately equal to ω of the fluid due to the discretization.

The discrete form of the velocity is therefore written as

$$\mathbf{u} \simeq \mathbf{u}^h = \sum_p \mathbf{K}[\mathbf{x} - \mathbf{x}_p(t)] \alpha_p(t). \quad (2.12)$$

Thus the discrete vorticity field is an N -body problem inducing velocity on each and implicitly evolving the vorticity field. This is one of the advantage of the vortex particle method as there are many ways to efficiently treat the problem. The N -body problem can be parallelized and can be accelerated using fast summation methods such as FMM, see 2.5.

However, like all N -body problem, equation 2.10 has a singularity when the particles approach each other and can result in numerical instability. To overcome this we can mollify the kernel, removing the singularity.

2.2.3 Convection of vortex blobs

In the discrete of the convection equation 2.6 of the viscous-splitting algorithm, the is solved as system of ODEs, where

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p), \quad (2.13)$$

with

$$\frac{d\alpha_p}{dt} = 0. \quad (2.14)$$

As the diffusion is done at the next sub-step, we have to ensure that the circulation is conserved.

2.2.4 Mollified vortex kernels

A vortex particle with a mollified core, non-zero core-size, is referred to as vortex blobs. The advantage of the vortex blobs is that the with a smooth distribution of the vorticity, the singularity disappears and so numerical instability does not happen when blobs get too close to each other. An ideal choice for a cutoff function is a Gaussian distribution, figure 2.2.

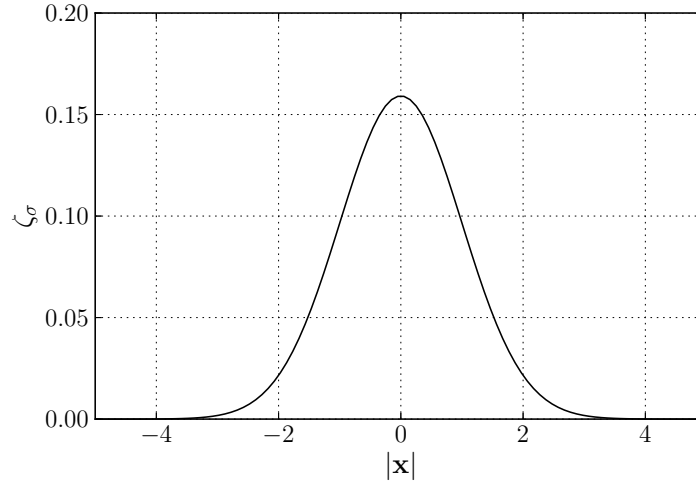


Figure 2.2: Vortex blob with Gaussian distribution: $[k = 2, \sigma = 1.0]$

Gaussian kernels satisfy the requirement for smooth distribution and decays quickly and is defined as

$$\zeta_\sigma = \frac{1}{k\pi\sigma^2} \exp\left(-\frac{|\mathbf{x}|}{k\sigma^2}\right), \quad (2.15)$$

where k is 1, 2 or 4 and determines the width of the kernel, σ is core-size of the blob. Note that smoothing function is chosen such that $\int \zeta = 1$, ensuring the conservation of circulation when mollified. So, using a smooth cut-off function ζ_σ , the mollified kernel \mathbf{K}_σ is given as

$$\mathbf{K}_\sigma = \mathbf{K} \star \zeta_\sigma. \quad (2.16)$$

The mollified vorticity field, represented by vortex blobs is given as

$$\omega^h(\mathbf{x}, t) = \sum_p \alpha_p(t) \zeta_\sigma[\mathbf{x} - \mathbf{x}_p(t)], \quad (2.17)$$

now representing the mollified vorticity field and equivalently, the mollified velocity field is given as

$$\mathbf{u}^h(\mathbf{x}, t) = \sum_p \mathbf{K}_\sigma[\mathbf{x} - \mathbf{x}_p(t)] \alpha_p(t). \quad (2.18)$$

Koumoutsakos and Chorin [5], have shown that for proper communication between the particle, the particle needs to overlap,

$$\text{overlap} = \frac{\sigma}{h}, \quad (2.19)$$

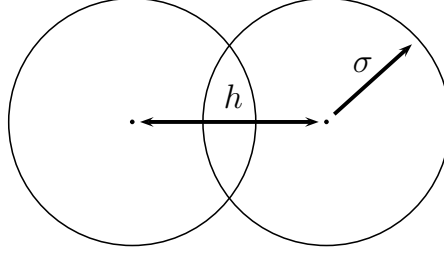


Figure 2.3: Vortex blob with overlap σ/h

where h is the nominal particle spacing, figure 2.3. If the particles fail to overlap, vortex blobs will also fail to recover the vorticity field. Such problems occurs when blobs are clustered due to high flow strain, leading to lagrangian grid distorting and must be treated, see section 2.2.6.

2.2.5 Vortex blob initialization

Now the question arises on how to initialize the particle's circulation strengths α_p . A common approach that is used is to estimate the particles strength is to say that

$$\alpha_p = \omega_p \cdot h^2. \quad (2.20)$$

This might seem like a valid assumption as the circulation of a given area is the integral of the vorticity in the area, equation 2.2, however this is no longer valid when regularizing the vorticity field using mollified gaussian kernels, equation 2.17. Barba and Rossi [1], has described this problem as gaussian blurring of the original vorticity field. Even though the particle have acquired the correct circulation strengths (i.e the local property), when evaluating the mollified vorticity field, we see that there is a mismatch in the evaluated vorticity field, figure 2.4.

Another way of viewing this characteristic is say the conservation of circulation is only valid globally, but not locally. A common standard for recovering the initial vorticity field is perform the Beale's method [2].

Beale's Iterative Method

The Beale's method is particle circulation processing scheme where the circulation of the particles are modified such that the mollified vorticity field matches the indented vorticity field. The recovery of the vorticity field is done by performing a discrete deconvolution,

$$\sum_j^N \beta_j \zeta_\sigma(\mathbf{x}_i - \mathbf{x}_j) = \omega_i, \quad (2.21)$$

where β_j is the circulation of the particles at positions \mathbf{x}_j such that it matches the exact vorticity ω_i at the position \mathbf{x}_i that we are evaluating. As we are try to solve for a N

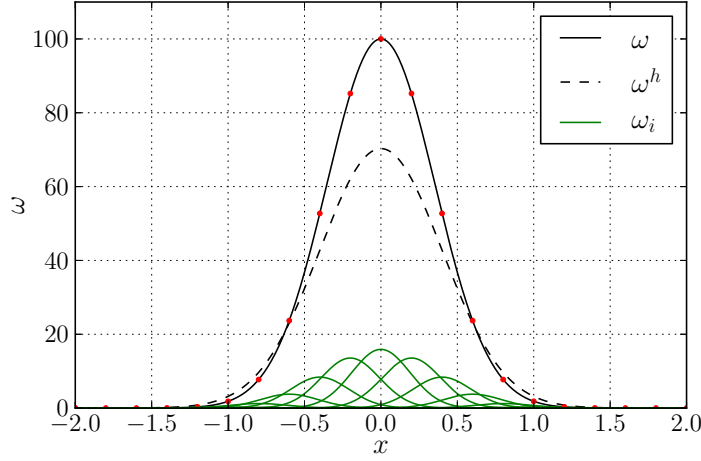


Figure 2.4: Mollified vorticity field of an arbitrary vorticity function with overlap = 1.0, $\sigma = 0.19$, $h = 0.19$. Vortex blob strength has been assigned by equation 2.20, sampling at exact vorticity [•, red dot]. Figure depicts exact vorticity distribution ω [—, solid], vorticity field of each blob ω_i [—, green dashed], the mollified vorticity field ω^h [- -, dashed].

unknown problem, we must set up a N system of equations. Multiplying both sides with the area associated to the blobs, we get

$$\mathbf{A}_{ij}\beta_j = \alpha_i^{\text{exact}}, \quad (2.22)$$

where

$$\mathbf{A}_{ij} = \zeta_\sigma(\mathbf{x}_i - \mathbf{x}_j) \cdot h^2 \quad (2.23)$$

is a $N \times N$ matrix containing the weights of the influence of each particle on each other. This matrix can be constructed by setting the Γ to one and determine the induced vorticity on each other. Furthermore, we see that it is not feasible to directly invert the matrix when we have large set of blobs but most importantly as the matrix \mathbf{A} is severely ill-conditioned [17], it should not be directly inverted. Beale's proposition to this problem was to iteratively solve for the solution,

$$\beta_j^{n+1} = \alpha_i + \beta_i^n - \mathbf{A}_{ij} \cdot \beta_j^n \quad (2.24)$$

We see that with just two iterations, the error between the mollified and exact vorticity field reduces drastically, figure 2.5. Koumoutsakos and Cottet [5], had shown that there was a drastic improvement in the velocity with just two to three iterations. However, we see that the cell vorticity of the blobs, directly evaluated from the particle strengths, equation 2.20, are more peaky and no longer matches the exact vorticity.

During the hybrid coupling algorithm, we see that this is the central source of coupling error between the eulerian and the lagrangian method, section ???. When performing the hybrid coupling, we need to recover the vorticity field transferred from the eulerian domain

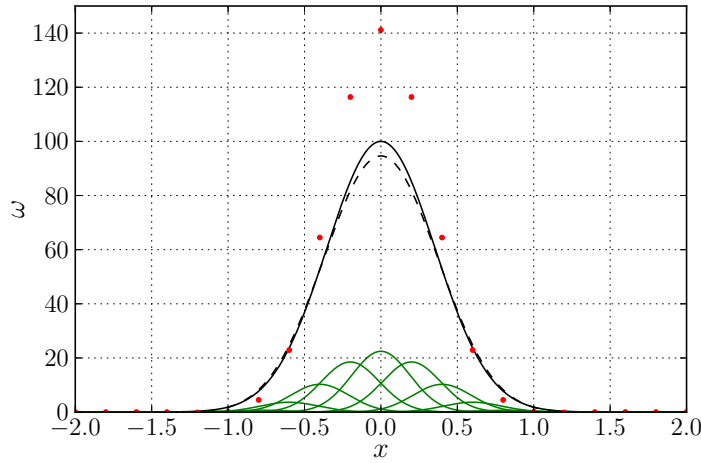


Figure 2.5: Mollified vorticity field after two Beale's iteration, overlap = 1.0, $\sigma = 0.19$, $h = 0.19$. Figure depicts exact vorticity distribution ω [—, solid], vorticity field of each blob ω_i [—, green dashed], the mollified vorticity field ω^h [- -, dashed].

to the lagrangian domain in every step. So, beale's correction is not a viable solution for the hybrid method. Thus there is a need for an alternate method of recovering the vorticity field.

!!! add the reference to hybrid !!!

Convergence of particle discretization

An alternate, temporary method to reduce the gaussian blurring of the vorticity field is to reduce the overlap (i.e. increase the overlap ratio) of the vortex blobs and to increase the spatial resolution.

Figure 2.6 shows mollified vorticity field results from modifying the spatial resolution parameters. Figure 2.6a shows the convergence of the mollified vorticity field ω^h to

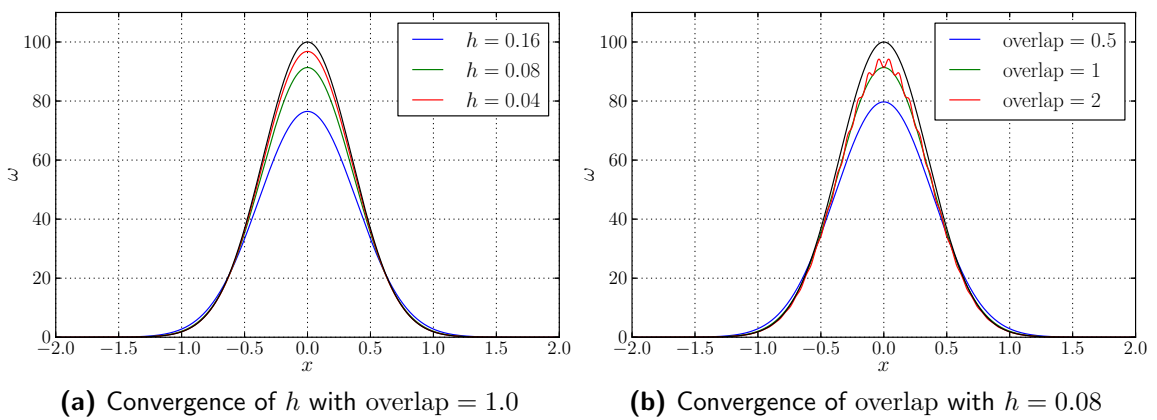


Figure 2.6: Convergence of vorticity by modifying the spatial resolution. Figure depicts exact vorticity field ω with [—, black] and various resolutions.

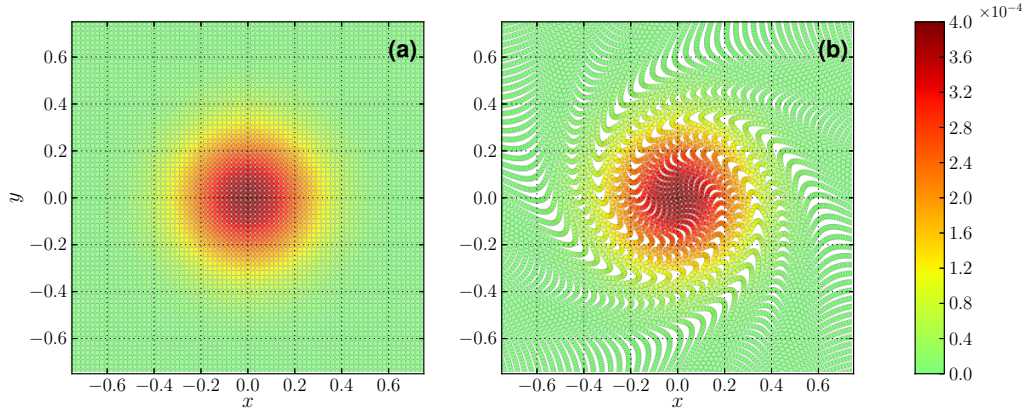


Figure 2.7: Lagrangian distortion of the vortex blobs after 100 steps. The initial vorticity field $\omega(\mathbf{x}, 0) = \exp(-12|\mathbf{x}|)$ with $\Delta t = 0.1$, $\sigma = 0.02$ and overlap = 1.0. Figure depicts (a) the initial and (b) the final distribution of the vortex blobs.

the exact vorticity field ω by reducing the nominal particle spacing h . The blobs have overlap = 1 and so the blob core-size σ is equal to h . We see that as you reduce the size of the blob and increase the number of particles, the mollified vorticity converges to the exact vorticity. Therefore, an alternate method of reducing the gaussian blurring is to increase the spatial resolution.

Furthermore, we could also adjust the overlap of the blobs, figure 2.6b. The σ and h of the blob is 0.08 and we see that increasing overlap number (i.e reducing the overlap), helps us to recover the original vorticity field. However, as explained by Koumoutsakos [5], if the overlap is too low, we lose the smooth recovery of the vorticity field. This is apparent when overlap = 2.0, where we see that the mollified vorticity field is fluctuation.

Therefore, for the hybrid coupling, we set overlap = 1.0 and maximize the spatial resolution at the coupling zone.

2.2.6 Remeshing scheme: Treating lagrangian grid distortion

During the convection step, we see that another source of error in the vorticity field is the lagrangian grid distortion. As we have seen before, when the vortex blobs fails to overlap, we are no longer able to reconstruct the correct the vorticity field, figure 2.6b. During the convection, due to the high strains in the fluid, the vortex blobs tent to clump together and creates regions where no vortex blobs are found, reducing the overlap of the blobs, figure 2.7.

We see that due to clustering of the vortex blobs, it fails to reproduce the correct vorticity field. A common strategy to overcome this problem is to remesh the vortex blobs to a uniform grid, so that we have a continuous vorticity field.

However, when transferring the vorticity from the old deformed grid to the new lagrangian uniform grid, we must satisfy the conservation laws of vorticity field. The interpolation methods is based on the conservation of linear impulse which directly implies the

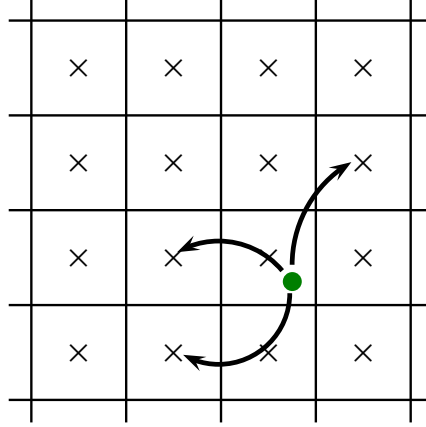


Figure 2.8: Remeshing of vortex blob (•, green) on the uniform grid defined by the (4×4) 2-D stencil.

conservation of the total circulation [5]. The transfer of the particle strengths is given as,

$$\alpha_p = \sum_q \tilde{\alpha}_q W \left(\frac{x_p - \tilde{x}_q}{h} \right), \quad (2.25)$$

where the strengths of the particles $\tilde{\alpha}_q$ of the distorted lagrangian grid \tilde{x}_q is transferred to the regular lagrangian grid x_p using the interpolation kernel, weighted W , giving us the remeshing particle strengths α_p . The transfer of strengths of one particles to it's interpolation nodes can be seen in figure 2.8.

M'_4 interpolation kernel

For lagrangian problem, we use the efficient interpolation kernel that has been used to reconstruct a smooth distribution interpolation, the M'_4 interpolation kernel, introduced by Monaghan [13]. In one dimension it is given as,

$$M'_4(\xi) = \begin{cases} 1 - \frac{5\xi^2}{2} + \frac{3|\xi|^3}{2} & |\xi| < 1, \\ \frac{1}{2}(2 - |\xi|)^2(1 - |\xi|) & 1 \leq |\xi| < 2, \\ 0 & 2 \leq |\xi|, \end{cases} \quad (2.26)$$

where

$$\xi = \frac{x_\nu - x_i}{h} \quad (2.27)$$

is the scale relative position of the particle x_ν to the i^{th} interpolation node x_i . The M'_4 is a third-order accurate piecewise smooth B-spline kernel, where $m = 4$ giving it 4 support nodes, figure 2.9. For the two dimensional problem that we have, the 2-D interpolation formula is simply tensor product of the 1-D interpolation kernel equation 2.26, and results in $4^2 = 16$ support nodes, figure 2.8.

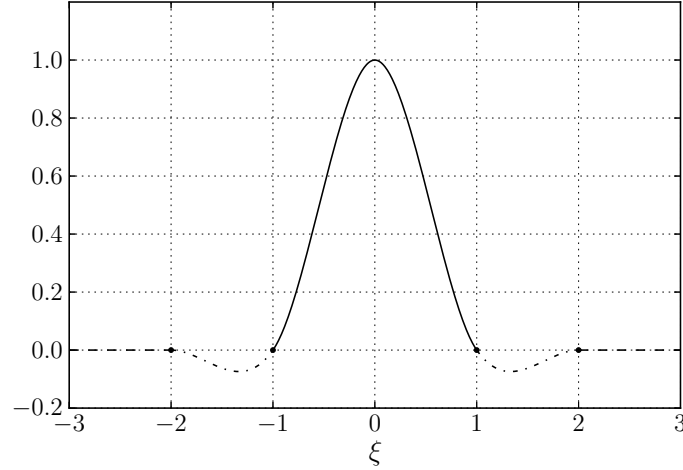


Figure 2.9: Monaghan's [13] third-order peicwise smooth B-spline kernel, M'_4

The interpolation kernel achieves the third-order accuracy as it also conserves the linear and the angular momentum of the vortex. Koumoutsakos [10] has investigated the drawback of the employing the remeshing strategy and have shown that there is approximately 4% decay in enstrophy of the flow due to sub-grid dissipation. Note that enstrophy \mathcal{E} , is defined as

$$\mathcal{E} = \frac{1}{2} \int_S \omega^2 dS \quad (2.28)$$

and is directly related to the kinetic energy of the fluid and gives an insight in the energy production and the dissipation of the fluid. Enstrophy is especially useful in turbulence flow investigation as it helps describe the energy cascade of the fluid.

2.3 Diffusion of Vortex Methods

So far, we have dealt with unbounded Euler flow, where we disregarded the viscosity of the flow. However, for real flow we must take in account of the diffusion of the vorticity. Chorin's approach to deal with the viscous term was to employ the viscous splitting algorithm. The flow is segregated to inviscid and viscous component and during the second sub-step we deal with the diffusion of the vorticity, equation 2.7. The equation can again be solved as a system of ODEs, similar to the convection step, where we say,

$$\frac{d\mathbf{x}_p}{dt} = 0, \quad (2.29)$$

with

$$\frac{d\alpha_p}{dt} = \nu \Delta \alpha_p. \quad (2.30)$$

Therefore in the diffusive step, we fix the position of the vortex blobs and only have to modify the strengths of the particles to mimic the diffusion process. Chorin initially

employed a random walk method, which generates and disperse vorticity using pseudo-random number algorithm [3]. However, this method suffers some limitations in accuracy. Since then Particle Strength Exchange (PSE) method [7], has been a common approach that has been used to treat diffusion.

Particle Strength Exchange

The Particle Strength Exchange method, first proposed by Mas-Galic [7], showed that diffusion can be treated for a particle method isotropic and anisotropic viscosity by approximating the diffusion operator (laplacian) with an integral operator and discretizing the operator using particles. The PSE can be seen as circulation correction method, where during the diffusion step of the viscous splitting algorithm, the strengths of the particle are corrected such that it accounts for the diffusion.

Vortex Redistribution Method

An alternative method to simulate the diffusion is to use the Vortex Redistribution Method (VRM) [14]. The model simulates diffusion by distributing the fraction of circulation of the vortex blobs to each other satisfying the diffusion. The model is based on conserving the moments of the particles by satisfying a linear system of equations. The circulation of the particle are transfer to the nearby particles that are

$$h_\nu = \sqrt{\nu \Delta t_d} \quad (2.31)$$

where h_ν is the diffusion distance and is directly related to the kinematic viscosity ν and the diffusive time-step Δt_d of the simulation. Not that the diffusive time-step Δt_d is equal to the convective time-step Δt_c if the diffusion process is done during every time-step. However, we can easily adjust the diffusion time-step and perform diffusion at a multiple step of the convection.

This is vital as a VRM (and also the PSE) requires a search algorithm to determine the particles that are within the zone of influence. A direct evaluation required $\mathcal{O}(N^2)$ evaluation, however can be speed up to $\mathcal{O}(\log N)$ using search tree algorithm.

The downside of the this approach, as also for the standard remeshing approach is the global remeshing generates large computation data and scales with the number of particles N . Therefore for problems with large number of particles, a tree-structured remeshing would be more feasible strategy [22].

2.3.1 Modified remeshing for treating diffusion

From further investigation of the VRM, we see that it is similar to remeshing strategy used to counter the lagrangian distortion during the convection process. Therefore Ghoniem and Wee [20] has proposed to combine the remeshing and the diffusion. The application of this methodology was validated by Speck [17]. The diffusion is simulated by the modifying the interpolation kernel of the remeshing process. The key advantage of the

modified remeshing method is that now it deals with the uniform remeshing grid helping us eliminating the computational expensive research requirement. The second advantage as shown by the authors is the simplicity. The method only requires a slight modification to the original remeshing tool.

During remeshing, the heat equation is satisfied by transferring the correct fraction of circulation to produce the proper amount of diffusion. The M'_4 kernel was modified to treat the diffusion and is given by:

$$M'_4(\xi, c) = \begin{cases} 1 - \frac{5\xi^2}{2} + \frac{3|\xi|^3}{2} - c^2(2 - 9\xi^2 + 6|\xi|^3) & |\xi| < 1, \\ \frac{1}{2}(2 - |\xi|)^2(1 - |\xi|) - c^2(2 - |\xi|)^2(1 - 2|\xi|) & 1 \leq |\xi| < 2, \\ 0 & 2 \leq |\xi|, \end{cases} \quad (2.32)$$

where

$$c^2 = \frac{\nu \Delta t_d}{h^2}, \quad (2.33)$$

and corresponds to the transfer quantity for diffusion. The addition terms in the interpolation kernel accounts for the diffusion process. When $c \rightarrow 0$, the interpolation kernel turns to the classical non-diffusion kernel.

Similar to the VRM, we can perform the remeshing at a given multiple step k_d of the convection step. However, we have an additional constraint on the diffusion time-step Δt_d , equation 2.33. Wee and Ghoniem [20] also investigated the error growth and the stability properties of the interpolation kernel in the Fourier space and have determined that for M'_4 interpolation kernel, we must satisfy

$$\frac{1}{6} \leq c^2 \leq \frac{1}{2}. \quad (2.34)$$

to ensure amplification factor and the phase error does not grow. This will ensure the stability of the problem and will suppress any spurious oscillations and ensure that it is a non-negative interpolation kernel with non-negative redistribution fractions.

However, we see that constraint on c^2 not only imposes a maximum Δt_d but also a constraint on the minimum Δt_d . This means that in scenarios, the $\Delta t_d = k_d \cdot \Delta t_c$ and this might introduce sudden change in vorticity field. Therefore, during the coupling of the lagrangian and eulerian domains, we would like to diffuse the vorticity after every convection scheme.

Simple redistribution scheme

The simple redistribution scheme based on the Shankar and Van Dommelen [14], developed by Tutty [18], makes it possible to remesh and diffuse the vorticity after every convection step,

$$\alpha_i^{n+1} = \sum_k \alpha_k^n W_{ki}^n, \quad (2.35)$$

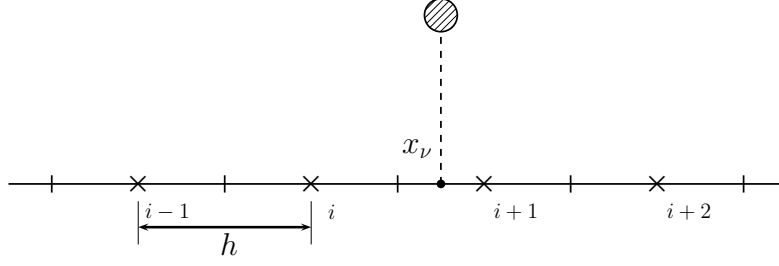


Figure 2.10: One dimensional, Simple redistribution scheme, diffusing the vortex blobs to the four stencil points ($k = i - 1, \dots, i + 2$) with vortex blob positioned at $x_i \leq x_\nu \leq x_{i+1}$ and nominal blob spacing h .

where W_{ki}^n is the fraction of circulation from vortex blob k transferred to vortex blob i by diffusion during the time step n . The fractions W_{ki}^n are calculated by conserving vorticity, center of vorticity, linear, and angular momentum of the vortex blobs [18]. In two dimensional problem the redistribution fractions are simple tensors products of the x, y one dimensional redistribution fractions,

$$W_{kl} = F_k G_l, \quad k = i - 1, \dots, i + 2, \quad l = j - 1, \dots, j + 2 \quad (2.36)$$

giving it a 16 point stencil. The one-dimension redistribution fractions for x -direction is a linear combination of the two basis solution of the redistribution equation for conservations,

$$F_k = (1 - \Delta) \cdot f_k + \Delta \cdot g_k, \quad k = i - 1, \dots, i + 2 \quad (2.37)$$

having a four point stencil, figure 2.10. The basis solution of redistribution are

$$f_i = 1 - 2 \left(\frac{h_\nu}{h} \right)^2 - \xi^2 \quad (2.38a)$$

$$f_{i-1} = \frac{1}{2} (1 - f_i - \xi) \quad (2.38b)$$

$$f_{i+1} = \frac{1}{2} (1 - f_i + \xi) \quad (2.38c)$$

and

$$g_{i+1} = 1 - 2 \left(\frac{h_\nu}{h} \right)^2 - \xi_1^2 \quad (2.39a)$$

$$g_i = \frac{1}{2} (1 - g_{i+1} - \xi_1) \quad (2.39b)$$

$$g_{i+2} = \frac{1}{2} (1 - g_{i+1} + \xi_1) \quad (2.39c)$$

where ξ is given by equation 2.27, $\xi_1 = \xi - 1$ are the distances between the k^{th} stencil nodes x_k and the vortex blob that is to be diffused with $x_i \leq x_\nu \leq x_{i+1}$. Note, f_k and

g_k is zero for all the other k . In the above equation, h_ν is the characteristic diffusion distance over the time Δt_d ,

$$h_\nu = \sqrt{\Delta t_d \cdot \nu}. \quad (2.40)$$

The only constraint imposed for positive redistribution fraction is

$$\frac{h_\nu}{h} < \frac{1}{\sqrt{2}}. \quad (2.41)$$

giving us the maximum time-step size constraint of

$$\Delta t_d < \frac{h^2}{2\nu}. \quad (2.42)$$

So now, we are able to perform diffusion, together with the convection step (i.e $k_d = 1$).

2.4 Boundary conditions at solid boundary

So far, we have only dealt with unbounded flow. During bounded flow simulation, we must impose addition constraint of the boundary to the simulation to simulate any flow about a geometry. From Helmholtz decomposition, we can have decompose the velocity field to the rotation and the irrotation component, equation 2.8. With the Helmholtz decomposition, we can use the potential component to prescribe the boundary conditions at the solid wall boundary,

$$\mathbf{u}_\phi = \nabla \Phi. \quad (2.43)$$

The incompressibility constraint results in a Laplace's equation for the potential field and unique solution is obtained by enforcing the wall boundary conditions,

$$\mathbf{u}_b \cdot \hat{\mathbf{n}} = (\mathbf{u}_w + \nabla \Phi) \cdot \hat{\mathbf{n}}, \quad (2.44)$$

and is defined as enforcing the no-through flow at the solid boundary wall, moving at \mathbf{u}_b . Note that the $\hat{\mathbf{n}}$ is the normal vector of the solid boundary. A classical approach for determine the solution to the Laplace's equation is by Green's function formulation. This approach required a singularity distribution over the body resulting in the appropriate boundary condition. Doublets and/or source panels are used to attain the required potential such that equation 2.44 is satisfied.

Linked boundary conditions

However Koumoutsakos, Leonard and Pepin [11], have suggested to enforce the boundary conditions through vortex sheets. The alternative approach of enforcing the solid boundary condition is not to decompose the velocity field into potential and rotational but to consider the solid boundary as an extension of the vorticity field through vortex sheets γ , figure 2.11. Due to the non-zero tangential velocity at the surface, a sudden

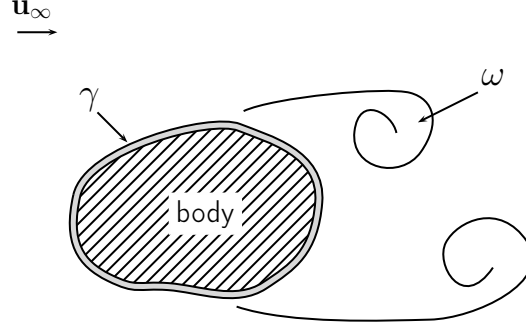


Figure 2.11: Extended vorticity field consisting of vorticity in the fluid and vortex sheet distribution around the body.

discontinuity in the velocity field can be considered as vortex sheet. Therefore, now to enforce the boundary conditions of the solid wall, we must satisfy the no slip velocity at the boundary,

$$\mathbf{u} \cdot \hat{\mathbf{s}} = \mathbf{u}_b \cdot \hat{\mathbf{s}}. \quad (2.45)$$

Koumoutsakos [12] only relies on the vortex sheet to enforce the no-slip velocity as no-slip boundary condition directly satisfies the no-through boundary conditions, also known as the linked boundary conditions. This was also proven by Shiels [15] and further investigated by Cooper, Mar and Barba [4] and shown that vortex sheet is the only necessary singularity element required to satisfy both the boundary conditions. Therefore, enforcing the no-slip boundary condition directly satisfies the no-through constraint at the surface.

2.4.1 Boundary integral equations

Thus the decomposed velocity field can be summarized as

$$\mathbf{u} = \mathbf{u}_\omega + \mathbf{u}_\gamma + \mathbf{u}_\infty \quad (2.46)$$

where the \mathbf{u}_γ denotes the velocity field induced by the vortex sheet. Applying the no-slip boundary conditions, we can say that

$$(\mathbf{u}_{\text{ext}} + \mathbf{u}_\gamma) \cdot \hat{\mathbf{s}} = \mathbf{u}_b \cdot \hat{\mathbf{s}} \quad (2.47)$$

where $\mathbf{u}_{\text{ext}} = \mathbf{u}_\omega + \mathbf{u}_\infty$ is the velocity field induced from domain external of the body and the vortex sheet. For the tangential boundary condition, for a well-conditioned system of equations, the induced velocity from the vortex sheet can be summarized as

$$(\mathbf{u}_{\text{ext}} - \mathbf{u}_b) \cdot \hat{\mathbf{s}} = \mathbf{u}_\gamma \cdot \hat{\mathbf{s}}. \quad (2.48)$$

The equation states that underneath the vortex sheet, we must have a vortex sheet inducing the velocity field \mathbf{u}_γ to counter the slip velocity $\mathbf{u}_{\text{slip}} = (\mathbf{u}_{\text{ext}} - \mathbf{u}_b)$ such that we satisfy the no-slip kinematic boundary conditions. Note that, \mathbf{u}_γ of equation 2.48, is negative of equation 2.47 as we looking under the vortex sheet. Koumoutsakos [12], expressed the

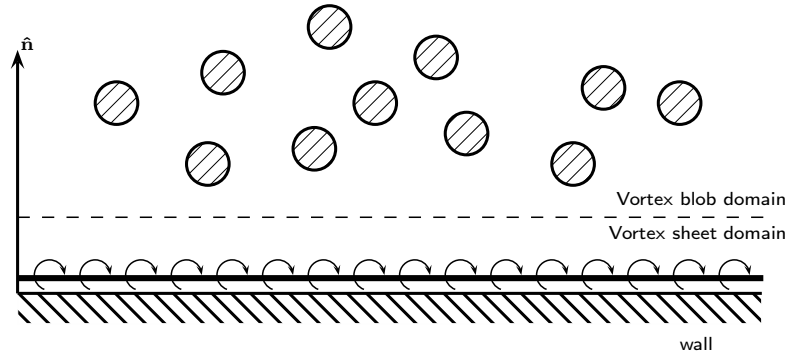


Figure 2.12: Extended vorticity field: Vortex sheet being an extension to the vorticity field (resolved by the vortex blobs) which is able to fully resolve the boundary vorticity

relation of the vortex sheet strengths to the no-slip boundary condition at the surface of the body (inside the body) through the Fredholm integral equation of the second kind,

$$-\frac{\gamma(s)}{2} + \frac{1}{2\pi} \oint \frac{\partial}{\partial n} [\log |\mathbf{x}(s) - \mathbf{x}(s')|] \gamma(s') ds' = \mathbf{u}_{\text{slip}} \cdot \hat{\mathbf{n}}. \quad (2.49)$$

The Left Hand Side (LHS) of the equation states at the point s , the velocity discontinuity is due to the vortex sheet at the point and the induced velocity of all the other vortex sheet acting on the point.

However, the equation 2.49 is not unique and accepts arbitrary solution for the vortex sheet strengths, therefore we must impose an additional constraint on the strength of the vortex sheet. The additional constraint is a constraint on the net circulation of the vortex sheet. As the vortex sheet is considered as an extension to the vorticity field, we could say that the net circulation of the vortex panels is the circulation that was not captured by the vortex blobs, figure 2.12. As vortex blobs are not available very-near the body, it is not possible for the vortex blobs to resolve the high gradient vorticity at the boundary. Therefore, all the vorticity in the vortex sheet domain must be captured by the vortex sheet.

When considering a moving boundary, we must also take into account of the circulation of the body due to its motion. The net circulation of the problem is given as

$$\Gamma = \Gamma_{\omega} + \Gamma_{\gamma} + \Gamma_b = 0, \quad (2.50)$$

where Γ_{ω} is the circulation of the vorticity in the fluid (i.e. the vortex blobs), Γ_{γ} is the wall bounded circulation (i.e. vortex sheets),

$$\Gamma_{\gamma} = \oint_S \gamma(s) ds, \quad (2.51)$$

and the Γ_b is the circulation of the moving body. The circulation inside the body can be considered also be considered as an extension to the vorticity field, where the body is filled with uniform vorticity due to the rotation of the body. Therefore the circulation of

a moving body can be calculated simply by integrating the “vorticity” inside the body

$$\Gamma_b = \iint_{body} \nabla \times \mathbf{u}_b \, dA. \quad (2.52)$$

So the constraint imposed on the net circulation of the vortex sheets is given as,

$$\Gamma_\gamma = -(\Gamma_\omega + \Gamma_b). \quad (2.53)$$

Now, in a pure lagrangian, we must transfer the vorticity generated from the body to the fluid. This is typically done by diffusing the vorticity of the vortex on the fluid, however in our hybrid coupling method, we can use the eulerian domain to introduce the vorticity into the fluid. The eulerian domain acts as the near-wall solver [6], and the strengths of the particles is interpolated from the eulerian domain, section ??.

!!! CiTe Hybrid sectin !!!

2.4.2 Panel method for treating no-slip boundary condition

Equation 2.49 is solved by discretizing the body into M vortex panels giving us a system of equations which can be used to determine the M unknowns of the strength of the vortex panels. The method referred to as the panel method has been greatly summarized by Katz and Plotkin [9].

Katz and Plotkin have shown several types of panel distributions with various orders of accuracy, from 0th order point vortex or up to 2nd order varying panel strength curved distribution. For this project, we have used a constant-strength vortex distribution that discretized the vortex sheet into straight segments, classified as the Constant-Strength Vortex Method (CSVM).

Panel methods are constructed by discretizing the integral equation and forming a system of equations to solve the M unknowns of the vortex panel,

$$\underbrace{\begin{pmatrix} -\frac{1}{2} & a_{12} & \cdots & a_{1M} \\ a_{21} & -\frac{1}{2} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & -\frac{1}{2} \end{pmatrix}}_{\mathbf{A}_{MM}} \underbrace{\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_M \end{pmatrix}}_{\vec{\gamma}} = \underbrace{\begin{pmatrix} \text{RHS}_1 \\ \text{RHS}_2 \\ \vdots \\ \text{RHS}_M \end{pmatrix}}_{\vec{\text{RHS}}} \quad (2.54)$$

where \mathbf{A}_{MM} contains the weights of the influence of the vortex panels $\vec{\gamma}$ on each other and the RHS is

$$\text{RHS} = \mathbf{u}_{\text{slip}} \cdot \hat{\mathbf{s}} \quad (2.55)$$

is the boundary condition to the system of equations. In addition, we have another constraint on the net circulation of the vortex panels and in discrete form we say that,

$$\sum_i^M \gamma_i \Delta s = \Gamma_\gamma \quad (2.56)$$

and results in a $M + 1$ system of equations for solving the M unknown strengths of the vortex panels,

$$\underbrace{\begin{pmatrix} -\frac{1}{2} & a_{12} & \cdots & a_{1M} \\ a_{21} & -\frac{1}{2} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & -\frac{1}{2} \\ \Delta s_1 & \Delta s_2 & \cdots & \Delta s_M \end{pmatrix}}_{\mathbf{B}_{(M+1)M}} \underbrace{\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_M \end{pmatrix}}_{\vec{\gamma}} = \underbrace{\begin{pmatrix} \text{RHS}_1 \\ \text{RHS}_2 \\ \vdots \\ \text{RHS}_M \\ \Gamma_\gamma \end{pmatrix}}_{\vec{\text{RHS}}} \quad (2.57)$$

However, as we have an additional constraint on the net strength of the panels, we have $M + 1$ set of equations with M unknowns giving us a overdetermined problem. The approach to solve such a problem is either by using a Least-Square solution method (LSTSQ), or introducing a new unknown or as used by Katz, eliminating an equation. Enforcing the no-slip boundary condition at each panel location is our vital criteria and therefore, elimination of an equation is not a viable strategy and so the LSTSQ method was utilized for the project.

Constant-Strength Vortex Method

The Constant-Strength vortex method (CSVM) is based on the flat (straight) discretization of the vortex sheet, where the panel have constant vortex strength as per definition. To solve the strengths of the panel problem, we enforce the dirichlet velocity boundary conditions at the collocation point x_{cp} , which is located just below the vortex sheet, figure 2.13b. The coefficient a_{ij} of the influence matrix \mathbf{A} is defined as

$$a_{ij} = \hat{\mathbf{u}}_{ij} \cdot \hat{\mathbf{t}}_i, \quad (2.58)$$

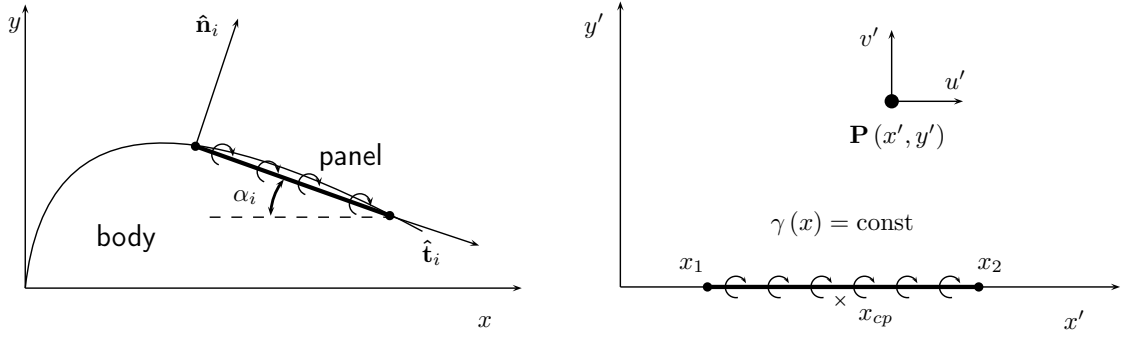
and is the tangential influence coefficient of j^{th} panel on the i^{th} panel. Therefore, induced velocity $\hat{\mathbf{u}}_{ij} = (\hat{u}, \hat{v})_{ij}$ is the unit induced velocity of the j^{th} panel on to the collocation points of i^{th} panel, where the vortex panels have unit strength (i.e $\hat{\gamma}_i = 1$). In general, the induced velocity of the vortex panels are calculated in the local panel coordinates, figure 2.13, and the transformation from the local panel coordinate (x', y') to the global coordinates systems (x, y) is given by

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} \cos \alpha_j & \sin \alpha_j \\ -\sin \alpha_j & \cos \alpha_j \end{bmatrix} \cdot \begin{bmatrix} u'_{ij} \\ v'_{ij} \end{bmatrix} \quad (2.59)$$

The induced velocity of the vortex panel j on the collocation point i is given as

$$u'_{ij} = \frac{\gamma_j}{2\pi} \left[\tan^{-1} \frac{y'_i - y'_{j,2}}{x'_i - x'_{j,2}} - \tan^{-1} \frac{y'_i - y'_{j,1}}{x'_i - x'_{j,1}} \right], \quad (2.60a)$$

$$v'_{ij} = -\frac{\gamma_j}{4\pi} \ln \frac{(x'_i - x'_{j,1})^2 + (y'_i - y'_{j,1})^2}{(x'_i - x'_{j,2})^2 + (y'_i - y'_{j,2})^2} \quad (2.60b)$$



(a) Panel discretization of the body in the global cartesian coordinates system (x, y) and local panel coordinates rotated by α_i . **(b)** Constant strength vortex panel in the local panel coordinate system (x', y') inducing velocity on the point P .

Figure 2.13: Vortex panel's global **(a)** and local **(b)** coordinates system definition as defined by Katz and Plotkins [9].

where $(x'_1, y'_1)_j$ and $(x'_2, y'_2)_j$ are the coordinates of the panel start and end point in its local panel coordinate system. Note that the self-induction of the vortex panel (when $i = j$), the influence coefficient becomes $a_{ij} = -1/2$, and can be seen in the diagonal terms of equation 2.54. So, by taking $\hat{\gamma}_i = 1$, we are able to construct the influence coefficients of the influence matrix \mathbf{A} and can solve the strengths of the panel,

$$\mathbf{B} \cdot \gamma = \text{RHS}, \quad (2.61)$$

using the LSTSQ method.

If we are dealing with multiple geometries, figure 2.14, the panel method can be easily

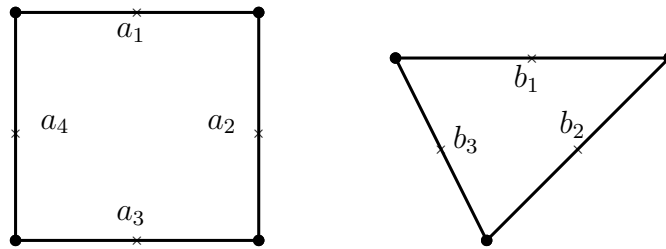


Figure 2.14: twoPanelBodies

extended by constructing a global influence matrix,

$$\underbrace{\begin{pmatrix} c_{a_1 a_1} & \cdots & c_{a_1 a_N} & c_{a_1 b_1} & \cdots & c_{a_1 b_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{a_N a_1} & \cdots & c_{a_N a_N} & c_{a_N b_1} & \cdots & c_{a_N b_M} \\ c_{b_1 a_1} & \cdots & c_{b_1 a_N} & c_{b_1 b_1} & \cdots & c_{b_1 b_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{b_M a_1} & \cdots & c_{b_M a_N} & c_{b_M b_1} & \cdots & c_{b_M b_M} \\ \Delta s_{a_1} & \cdots & \Delta s_{a_N} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \Delta s_{b_1} & \cdots & \Delta s_{b_M} \end{pmatrix}}_{\begin{pmatrix} AA & AB \\ BA & BB \\ \Delta s_A & 0 \\ 0 & \Delta s_B \end{pmatrix}} \begin{pmatrix} \gamma_{a_1} \\ \vdots \\ \gamma_{a_N} \\ \gamma_{b_1} \\ \vdots \\ \gamma_{b_M} \end{pmatrix} = \underbrace{\begin{pmatrix} \text{RHS}_{a_1} \\ \vdots \\ \text{RHS}_{a_N} \\ \text{RHS}_{b_1} \\ \vdots \\ \text{RHS}_{b_M} \\ \Gamma_{\gamma,a} \\ \Gamma_{\gamma,b} \end{pmatrix}}_{\begin{pmatrix} \text{RHS}_a \\ \text{RHS}_b \\ \Gamma_{\gamma,a} \\ \Gamma_{\gamma,b} \end{pmatrix}} \quad (2.62)$$

where the diagonal matrices (AA, BB) are the self-induction matrix of the panel body and the non-diagonal terms (AB, BA) are the inter-induction matrix containing the panel influence of $B \rightarrow A$ and $A \rightarrow B$ respectively. The final two rows of the induction matrix contains the additional circulation constraint for each body.

2.4.3 Convergence study of panel method

2.5 Simulation acceleration techniques

2.5.1 Fast multi-pole Method

2.5.2 Parallel computation in GPU

2.6 Validation of lagrangian method

2.6.1 Lamb-oseen vortex at $Re = 100$

2.6.2 Convergence study of the viscous vortex method

2.6.3 Convection of Clercx-Bruneau dipole at $Re = 625$

2.7 Summary

Chapter 3

Eulerian Domain: Finite Element Method

3.1 Introduction to Finite Element Method

3.1.1 Finite element discretization

3.1.2 Finite element function and function space

3.2 Solving the Finite Element problem

3.2.1 Introduction to FEniCS Project

3.2.2 Mesh generation using GMSH

3.3 Solving Incompressible Navier-Stokes Equations

3.3.1 Velocity-pressure formulation

3.3.2 Incremental pressure correction scheme

3.3.3 Determining the vorticity field

3.3.4 Determining the body forces

Frictional Forces

Pressure Forces

3.4 Validation of eulerian method

3.4.1 Clercx-Bruneau dipole collision at $Re = 625$

3.4.2 Impulsively started cylinder at $Re = 550$

3.5 Summary

Hybrid Eulerian-Lagrangian Vortex Particle Method

4.1 Theory of Domain Decomposition Method

4.1.1 Advantage of domain decomposition

4.1.2 Assumptions and Limitations

4.1.3 Modified coupling strategy

4.2 Eulerian-Lagrangian coupling algorithm

4.2.1 Eulerian dirichlet boundary condition

4.2.2 Vorticity interpolation algorithm

4.3 Introduction to pHyFlow: Hybrid solver

4.3.1 Program structure

4.4 Summary

Chapter 5

Verification and Validation of Hybrid Method

5.1 Error in coupling: Verification with Lamb-Ossen vortex

5.1.1 Generation of artificial vorticity

5.2 Clercx-Bruneau dipole convection at $Re = 625$

5.2.1 Comparison of vorticity contours

5.2.2 Variation in maximum vorticity

5.2.3 Variation in kinetic energy

5.2.4 Variation in enstrophy

5.3 Clercx-Bruneau dipole collision at $Re = 625$

5.3.1 Comparison of vorticity contours

5.3.2 Variation in maximum vorticity

5.3.3 Variation in kinetic energy

5.3.4 Variation in enstrophy

5.3.5 Variation in palinstrophy

5.4 Impulsively started cylinder problem at $Re = 550$

5.4.1 Evolution of the wake

5.4.2 Evolution of pressure and friction drag

5.4.3 Evolution of lift

5.5 Moving body

Conclusion and Recommendation

6.1 Conclusion

6.1.1 Lagrangian domain

6.1.2 Eulerian domain

6.1.3 Hybrid method

6.2 Recommendations

6.2.1 Lagrangian domain

6.2.2 Eulerian domain

6.2.3 Hybrid method

References

- [1] L.a. Barba and Louis F. Rossi. Global field interpolation for particle methods. *Journal of Computational Physics*, 229(4):1292–1310, February 2010.
- [2] J.Thomas Beale. On the Accuracy of Vortex Methods at Large Times. In Bjorn Engquist, Andrew Majda, and Mitchell Luskin, editors, *Computational Fluid Dynamics and Reacting Gas Flows SE - 2*, volume 12 of *The IMA Volumes in Mathematics and Its Applications*, pages 19–32. Springer New York, 1988.
- [3] AJ Chorin. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*, 1973.
- [4] Christopher D Cooper, Santa Mar, and Lorena Barba. Panel-Free Boundary Conditions for Viscous Vortex Methods. *19th AIAA Computational Fluid Dynamics*, 2009(June):1–12, June 2009.
- [5] G H Cottet and P D Koumoutsakos. *Vortex Methods: Theory and Practice*, volume 12. Cambridge University Press, 2000.
- [6] G. Daeninck. *D EVELOPMENTS IN HYBRID APPROACHES Vortex method with known separation location Vortex method with near-wall Eulerian solver RANS-LES coupling*. PhD thesis, Universit catholique de Louvain, 2006.
- [7] S. Degond, P.; Mas-Gallic, Pierre Degond, and S Mas-Gallic. The weighted particle method for convection-diffusion equations. I. The case of an isotropic viscosity. *Mathematics of Computation*, 53(188):485–507, 1989.
- [8] CJ Simão Ferreira. The near wake of the VAWT: 2D and 3D views of the VAWT aerodynamics. 2009.
- [9] Joseph Katz and A Plotkin. *Low-speed aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, 2001.
- [10] P Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex. *Journal of Computational Physics*, 138(2):821–857, 1997.

-
- [11] P Koumoutsakos, A Leonard, and F Pépin. Boundary Conditions for Viscous Vortex Methods. *Journal of Computational Physics*, 113(1):52–61, July 1994.
 - [12] PD Koumoutsakos. Direct numerical simulations of unsteady separated flows using vortex methods. 1993.
 - [13] J.J Monaghan. Extrapolating B-splines for interpolation. *Journal of Computational Physics*, 60(2):253–262, September 1985.
 - [14] S. Shankar and L.van Dommelen. A New Diffusion Procedure for Vortex Methods. *Journal of Computational Physics*, 127(1):88–109, August 1996.
 - [15] Doug Shiels. *Simulation of controlled bluff body flow with a viscous vortex method*. Dissertation (ph.d.), California Institute of Technology, 1998.
 - [16] Carlos Simão Ferreira, Gijs Kuik, Gerard Bussel, and Fulvio Scarano. Visualization by PIV of dynamic stall on a vertical axis wind turbine. *Experiments in Fluids*, 46(1):97–108, August 2008.
 - [17] Robert Speck. *Generalized algebraic kernels and multipole expansions for massively parallel vortex particle methods*, volume 7. Forschungszentrum Jülich, 2011.
 - [18] O.R. Tutty. A Simple Redistribution Vortex Method (with Accurate Body Forces). *ArXiv e-prints*, September 2010.
 - [19] L.J. Vermeer, J.N. Sørensen, and a. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39(6-7):467–510, August 2003.
 - [20] Daehyun Wee and Ahmed F. Ghoniem. Modified interpolation kernels for treating diffusion and remeshing in vortex methods. *Journal of Computational Physics*, 213(1):239–263, March 2006.
 - [21] Wikipedia. Vertical-Axis Wind Turbine, July 2013.
 - [22] G S Winckelmans, J K Salmon, Universite Catholique De Louvain, M S Warren, A Leonard, Theoretical Astrophysics, Los Alamos National Laboratories, Los Alamos, B Jodoin, and Universite De Sherbrooke. Application of Fast Parallel and Sequential Tree Codes to Computing Three-Dimensional Flows with the Vortex Element and Boundary Element Methods. 1:225–240, 1996.