



TÚLIO RODARTE RICCIARDI

*Fast Multipole Discrete Vortex Method Applied  
to Unsteady Flow Simulations*

*Método de Vórtices Discretos e Multipolos  
Rápidos Aplicados em Escoamentos  
não-Estacionários*

CAMPINAS

2016

TÚLIO RODARTE RICCIARDI

*Fast Multipole Discrete Vortex Method Applied  
to Unsteady Flow Simulations*

*Método de Vórtices Discretos e Multipolos  
Rápidos Aplicados em Escoamentos  
não-Estacionários*

Dissertation presented to the School of Mechanical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Mechanical Engineering, in the area of Thermal and Fluids.

Orientador: William Roberto Wolf

Coorientador: Alex Mendonça Bimbato

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL  
DA DISSERTAÇÃO DEFENDIDA PELO ALUNO TÚLIO  
RODARTE RICCIARDI, E ORIENTADA PELO PROF.  
DR. WILLIAM ROBERTO WOLF.

Assinatura do Orientador

  
\_\_\_\_\_

Campinas  
2016

**Agência(s) de fomento e nº(s) de processo(s):** CAPES, 33003017

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Luciana Pietrosanto Milla - CRB 8/8129

R358f Ricciardi, Túlio Rodarte, 1991-  
Fast multipole discrete vortex method applied to unsteady flow simulations /  
Túlio Rodarte Ricciardi. – Campinas, SP : [s.n.], 2016.

Orientador: William Roberto Wolf.  
Coorientador: Alex Mendonça Bimbato.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade  
de Engenharia Mecânica.

1. Dinâmica de vórtices. 2. Mecânicas dos fluídos. 3. Interação partícula-  
partícula. I. Wolf, William Roberto, 1980-. II. Bimbato, Alex Mendonça. III.  
Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. IV.  
Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Método de vórtices discretos e multipolos rápidos aplicados em escoamentos não-estacionários

**Palavras-chave em inglês:**

Vortex dynamics

Fluid mechanics

Particle-particle interaction

**Área de concentração:** Térmica e Fluídos

**Titulação:** Mestre em Engenharia Mecânica

**Banca examinadora:**

William Roberto Wolf [Orientador]

Rafael dos Santos Gioria

Luiz Otávio Saraiva Ferreira

**Data de defesa:** 08-07-2016

**Programa de Pós-Graduação:** Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA  
DEPARTAMENTO DE TERMICA E FLUIDOS

DISSERTAÇÃO DE MESTRADO ACADÊMICO

*Método de Vórtices Discretos e Multipolos  
Rápidos Aplicados em Escoamentos  
não-Estacionários*

Autor: Túlio Rodarte Ricciardi

Orientador: William Roberto Wolf

Coorientador: Alex Mendonça Bimbato

A Banca Examinadora composta pelos membros abaixo aprovou esta dissertação:

**Prof. Dr. William Roberto Wolf**

**Instituição UNICAMP**

**Prof. Dr. Rafael dos Santos Gioria**

**Instituição USP**

**Prof. Dr. Luiz Otávio Saraiva Ferreira**

**Instituição UNICAMP**

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

Campinas, 08 de Julho de 2016.

## Acknowledgments

Agradeço à minha família pelo apoio incondicional.

Agradeço aos amigos do laboratório por todos os dias juntos.

Agradeço aos amigos de Brasília por todos esses anos de longas amizades, em especial à amizade incondicional do Daniel e de Laísa. Se cuida, Lai.

Agradeço ao Alex pela coorientação, ajudando sempre mesmo que de longe.

Agradeço ao William pela dedicação e paciência, por ter sempre 5 minutinhos quando temos dúvidas.

*No.*  
*Not even in the face of Armageddon.*  
*Never compromise.*  
*Rorschach*

# Abstract

The Discrete Vortex Method (DVM) is a meshless numerical method based on a Lagrangian description of the vorticity transport equation, which is split into diffusive and convective effects. In order to solve this equation, the vorticity field is discretized in  $N$  vortex-particles. Several formulations can be used to model the diffusive effects, e.g., the random walk method, the core spreading method, the particle strength exchange, etc. The convection term can be treated using a material derivative to avoid the solution of a non-linear term. Therefore, each vortex is convected with the fluid velocity field, which is evaluated by the contributions from the incident flow, the perturbation due to the body, and the vortex-vortex interactions calculated by the Biot-Savart law. However, the last contribution requires an expensive convolution step of  $O(N^2)$  calculations, which imposes a heavy limitation on the usage of the method to solve engineering problems. With that in mind, alternative ways are required to accelerate the DVM simulations.

The Fast Multipole Method is listed as one of the top 10 algorithms of the twentieth century, and it was developed by Greengard and Rokhlin (1987) for the solution of  $N$ -body gravitational problems. The algorithm consists of clustering the influence of elements close to each other, and then evaluating their interaction at distant locations, *i.e.*, the center of far away clusters, with computational cost  $O(N)$  for a large number  $N$ . This way, the influence among different groups of particles is computed faster than the  $O(N^2)$  operations required by the direct Biot-Savart law.

Here, we use the non-adaptive multi-level FMM with an optimization in the pre-processing steps, along with several techniques to speed up both pre-processing and FMM steps. The coupling of DVM and FMM is investigated in the present work, in three different problems: the simulation of the flow past an impulsively started cylinder and the temporal evolution of both an aircraft wake as well as a mixing layer. For these problems, there is a comparison of the computational time used by both the DVM-FMM and solely by the DVM.

Finally, as the temporal evolution of a mixing layer requires periodic boundary conditions, a solution of an alternative kernel for the FMM is also employed in order to solve the problem, followed by the investigation of its precision.

**Keywords:** fast multipole method, discrete vortex method, vortex dynamics

# Resumo

O método de vórtices discretos (DVM) não necessita de malhas por ser uma descrição lagrangiana da equação do transporte de vorticidade. Esta, por sua vez, é separada em termos difusivos e convectivos. Esta equação é resolvida pela discretização do campo de vorticidade em  $N$  vórtices discretos. Diversos métodos podem ser usados para modelar os efeitos da difusão; pode-se citar o método do passo aleatório, método do crescimento do núcleo, método da troca de intensidade, entre outros. Os termos de convecção são resolvidos pela utilização da derivada material para evitar termos não-lineares. Assim, cada vórtice discreto é convectado pelo campo local de velocidade, que é calculado pela contribuição do escoamento livre, superfícies sólidas e pela solução da lei de Biot-Savart que rege a interação entre vórtices. Entretanto, esta última contribuição exige um dispendioso passo de convolução com  $O(N^2)$  operações, que impõe restrição no uso do método para a solução de problemas típicos de engenharia. Assim, métodos alternativos são necessários para acelerar a solução do DVM.

O método de multipolos rápidos, FMM, considerado um dos 10 melhores algoritmos do século 20, foi proposto por Greengard and Rokhlin (1987) para a solução da interação gravitacional entre  $N$  corpos. O algoritmo consiste no agrupamento da influência de elementos próximo entre si, e então calcula-se a interação em regiões distantes, como por exemplo o centro de outro agrupamento. Esta operação tem custo computacional de ordem  $O(N)$  para um número  $N$  suficientemente grande. Assim, a influência entre grupos distantes de elementos é calculada mais eficientemente do que a ordem  $O(N^2)$  para calcular diretamente a lei de Biot-Savart.

Neste trabalho, nós usamos um esquema não-adaptativo multi-nível do FMM com melhorias para acelerar o pré-processamento bem como os cálculos de interação no FMM. O acoplamento dos métodos é investigado para três diferentes problemas: a simulação de um cilindro abruptamente acelerado e a evolução temporal de uma esteira de aeronave assim como de uma camada de mistura. Uma comparação do custo computacional do método acelerado é comparado com a solução usando apenas a lei de Biot-Savart.

Finalmente, como uma camada de mistura requer condições de contorno periódicas, o estudo de uma série alternativa para o cerne do FMM é feito com a investigação da precisão e do tempo computacional.

**Palavras-chave:** método de multipolos rápidos, método de vórtices discretos, dinâmica de vórtices.

# List of Figures

1.1	Applications of the fast multipole method - extracted from Wolf (2011). . .	27
2.1	Coordinate system for BEM. . . . .	35
2.2	Representation of vortex interaction in Argand diagram for complex numbers. . . . .	36
2.3	Vorticity distribution in a Lamb-Oseen Vortex. . . . .	38
2.4	Self-similar velocity profiles for different kinds of vortex models. . . . .	39
2.5	Decay of the smoothing parameter. . . . .	40
2.6	Analysis of the time marching schemes. . . . .	44
2.7	Analysis of the invariants. . . . .	44
2.8	Convergence of diffusion process using the random walk method. . . . .	46
2.9	Diffusion for different $Re$ using 10k vortex particles. . . . .	46
3.1	Power series expansion of example function $f(x) = e^{-1/x^2}$ . . . . .	51
3.2	Interactions in the near-field and far-field. . . . .	52
3.3	Different types of refinement in the FMM. . . . .	53
3.4	Refinement of initial domain from level $\ell = 0$ to $\ell = 2$ . . . . .	54
3.5	Representation of interaction and neighbor lists for different levels in FMM. . . . .	55
3.6	Box creation in preprocessing. . . . .	58
3.7	Upward pass schematics. . . . .	58
3.8	Downward pass schematics. . . . .	59
3.9	Near-field calculations schematics. . . . .	59
3.10	Multi-level interactions in the far-field (in light gray). . . . .	60
3.11	Motion of vortex particles and changes in adaptive refinement. . . . .	63
3.12	Efficient way for mapping the location of vortex elements. . . . .	65
3.13	Comparison between the Lamb-Oseen and vortex blob models. . . . .	66
3.14	Details on the use of Lamb-Oseen vortex model applied to the FMM. . . . .	67
3.15	Spurious perturbations appearing at refinement level 4 and $\sigma = 0.05$ . . . . .	67
3.16	FMM domain refinement during step 20 to step 21. . . . .	69
3.17	FMM domain growth due to the escape of particles. . . . .	70
3.18	Computational time for the flow around a circular cylinder. . . . .	70
3.19	Development of circular cylinder wake. . . . .	71
3.20	Aerodynamic coefficients for the circular cylinder. . . . .	71
3.21	Lift spectrum showing the Strouhal number for the cylinder vortex shedding. . . . .	72

4.1	Vortical structures formed by aerodynamic wake from wing tip and flap. . .	78
4.2	Typical airflow vertical velocity distribution in a cross-flow plane behind an aircraft (extracted from Ginevsky and Zhelannikov (2009)). . . . .	78
4.3	Model of aircraft wake in Trefftz plane (extracted from Drela (2014)) . . .	79
4.4	Temporal evolution of the Trefftz plane problem with pathlines for different particles given by colored lines. . . . .	79
4.5	Rate of change in circulation along then span of an elliptic wing. . . . .	80
4.6	Different initial positions of the discrete vortex particles. . . . .	81
4.7	Induced vertical velocity for Lamb and vortex blob models. . . . .	82
4.8	Effects of spatial distribution of vortex-particles in the wake discretization. .	84
4.9	Error analysis of FMM at refinement level 2 due to different time steps. . .	85
4.10	Error analysis of FMM at different levels for time step $\Delta t = 1 \times 10^{-4}$ . . . .	85
4.11	Comparison of vortex models in the sheet roll-up. Results are obtained with $\delta = \sigma = 0.001$ , RK4, $\Delta t = 1 \times 10^{-5}$ , $t^* = 0.05$ , 25k particles and $n = 2$ . .	86
4.12	Comparison of time marching schemes for $\sigma = 0.001$ , $\Delta t = 1 \times 10^{-4}$ , $t^* =$ $0.05$ , $n = 2$ , and 25k vortex particles. . . . .	87
4.13	Time step effect for $\sigma = 0.001$ , RK4, $\Delta t = 1 \times 10^{-5}$ , $t^* = 0.045$ , $n = 2$ and 25k vortex particles. . . . .	87
4.14	Error from FMM to Biot-Savart direct summation. . . . .	88
4.15	Long-term truncation error for FMM and Biot-Savart law. . . . .	89
4.16	Discretization effects for different number of particles with $\sigma = 0.001$ , RK4, $\Delta t = 1 \times 10^{-5}$ , $t^* = 0.045$ and $n = 2$ . . . . .	89
4.17	Roll-up process for different viscous cores, $\sigma$ . Solutions obtained with 200k vortex particles, RK4, $\Delta t = 4 \times 10^{-7}$ , $t^* = 3 \times 10^{-5}$ and $n = 3$ . . . . .	90
4.18	Time step influence on roll-up process for 200k vortex particles, $\sigma = 1 \times 10^{-5}$ , RK4, $t^* = 3 \times 10^{-5}$ , $n = 3$ and Gaussian quadrature with 3 points. . . . .	91
4.19	Number of particles influence for $\sigma = 1 \times 10^{-5}$ , RK4, $\Delta t = 8 \times 10^{-8}$ , $t^* =$ $3 \times 10^{-5}$ , $n = 3$ and Gaussian quadrature with 3 points. . . . .	92
4.20	Influence of machine precision on suppression of instabilities for 200k vortex, $\sigma = 1 \times 10^{-5}$ , RK4, $\Delta t = 8 \times 10^{-8}$ , $t^* = 3 \times 10^{-5}$ , $n = 3$ and quadrature with 3 points. . . . .	93
4.21	Viscosity influence for 1000k vortex, $\sigma = 1 \times 10^{-5}$ , RK4, $\Delta t = 8 \times 10^{-8}$ , $t^* = 3.6 \times 10^{-5}$ , $n = 3$ and Gaussian quadrature with 3 points. . . . .	93
4.22	Computational time to evaluate the velocity field for Biot-Savart and FMM. . .	94
4.23	Computational time to evaluate far-field and near-field operations. . . . .	95
4.24	Influence of quadruple precision for both Biot-Savart and FMM computational time per step. . . . .	96

5.1	Kelvin-Helmholtz instability present on mixing-layers. . . . .	97
5.2	Sinusoidal initial perturbation, with amplitude $A = 0.01$ , of a shear layer. . . . .	98
5.3	Schematic of operations performed in the periodic FMM. . . . .	102
5.4	Optimized scheme for far-field interactions of periodic domains. . . . .	104
5.5	Velocity profile for periodic vortex models. . . . .	107
5.6	Relative error for both periodic vortex models. . . . .	108
5.7	Absolute error for both periodic vortex models. . . . .	108
5.8	Results for Biot-Savart with $\sigma = \delta = 0.05$ , 5120 vortex, RK4, $\Delta t = 0.01$ , $t^* = 2.0$ , and $A = 0.01$ . . . . .	109
5.9	Convergence study for the position of 5120 vortex with $\sigma = 0.02$ at $t^* = 2.0$ for different time steps. . . . .	110
5.10	Effect of number of vortex using Biot-Savart with RK4, $\Delta t = 0.01$ , and $A = 0.01$ . . . . .	111
5.11	Effect of initial perturbation for 2k vortex with $\sigma = 0.01$ using Biot-Savart, RK4 and $\Delta t = 0.01$ . . . . .	112
5.12	Effect of machine precision for 2k vortex with $\sigma = 0.01$ , $A = 0.01$ , Biot-Savart, RK4 and $\Delta t = 0.01$ . . . . .	112
5.13	Combination of smaller amplitude $A = 0.005$ and increased machine precision (16 bytes) with $\sigma = 0.01$ , Biot-Savart, RK4 and $\Delta t = 0.01$ . . . . .	113
5.14	Discretization of a finite shear layer using $N$ vortex elements. . . . .	114
5.15	Precession velocity for a shear layer discretized with 5120 vortex particles according to the number of periodic domains. . . . .	116
5.16	Influence of number of domains in periodicity for a finite shear layer discretized by 5120 vortex particles, $A = 0.01$ , $\sigma = 0.05$ , RK4 with $\Delta t = 0.01$ until $t^* = 2.0$ . . . . .	117
5.17	Error reduction when using a balanced shear layer. . . . .	118
5.18	Current implementation of the wake balance procedure using the FMM. . . . .	119
5.19	Investigation of the number of domains in periodic FMM using 5120 vortex, $A = 0.005$ , $\sigma = 0.02$ , RK4 with $\Delta t = 0.01$ until $t^* = 2.0$ . . . . .	120
5.20	Long-time integration error for 5120 vortex, $A = 0.005$ , $\sigma = 0.02$ , RK4 with $\Delta t = 0.01$ until $t^* = 2.0$ . . . . .	120
5.21	Temporal evolution of shear-layer discretized by 50k elements with $\sigma = 0.01$ convected by RK4 with $\Delta t = 0.01$ to $t^* = 2.0$ using periodic FMM and quadruple precision. . . . .	121
5.22	Detail view on the core region for long-time integration. . . . .	121
5.23	Computational time for a single evaluation of all convection steps. . . . .	122
5.24	Computational time for a single evaluation of periodic images in FMM. . . . .	123

6.1	Convergence properties of the first cotangent series. . . . .	125
6.2	Absolute value of the cotangent power series expansion based on the argument $z$ for several truncation terms $p$ evaluated with double precision. . . . .	127
6.3	Error of the cotangent power series expansion based on the argument $z$ for several truncation terms. . . . .	127
6.4	Convergence of the cotangent power series expansion. . . . .	128
6.5	Error of the fast exponential series summation for different truncation terms. . . . .	131
6.6	Error in velocity magnitude for a shear-layer, discretized by different numbers of vortices for double (dashed lines) and quadruple (solid) machine precision. . . . .	134
6.7	Light gray boxes in the exponential interaction list of that in dark gray. . . . .	136
6.8	Light gray boxes in the power-series interaction list of that in dark gray. . . . .	136
6.9	Interaction list (light gray) of a box (dark gray). . . . .	137
6.10	Refinement around the observer box for faster power-series solution. . . . .	138
6.11	Magnitude of the velocity field. . . . .	140
6.12	Error in velocity magnitude of the proposed fast algorithm compared to the Biot-Savart law computed using machine quadruple precision. . . . .	140
6.13	Components of the velocity field evaluated by the fast summation method. . . . .	140
6.14	Vorticity field of a random distribution of vortex particles. . . . .	141
6.15	Error for a single evaluation of local velocity field for different methods. . . . .	141
6.16	Overall computational cost of the combined method compared to the Biot-Savart solution. . . . .	142
6.17	Computational time for box-to-box operations. . . . .	143
6.18	Computational cost of steps linearly dependent on the number of elements. . . . .	144
6.19	Computational cost of the three algorithms employed for fast summation of the cotangent function. . . . .	145

## List of Tables

3.1	Comparison of aerodynamic coefficients for FMM-DVM, DVM and experimental data. . . . .	69
5.1	Accumulated error for different time steps, compared to $\Delta t = 0.001$ , to march the system to $t^* = 2.0$ . . . . .	110

# List of Acronyms

## Latin Letters

$A$	sinusoidal perturbation amplitude
$b$	FMM box
$B$	FMM box at the upper level of $b$
$B_{2n}$	Bernoulli number
$E$	absolute error
$G$	Green's function
$i$	imaginary number
$i$	periodic level
$I$	maximum periodic level
$\ell$	FMM refinement level
$L$	maximum refinement level of central domain
$n$	auxiliary exponent for vortices distribution
$n$	number of domains in periodic problems
$N$	total number of vortex-particles
$p$	truncation of Green's function series expansions
$r$	radial distance
$S$	FMM box size
$t$	time
$\mathbf{u}$	vector velocity
$u$	velocity component along $x$ -axis
$v$	velocity component along $y$ -axis
$w$	velocity on complex plane
$\mathbf{x}$	vector position
$x$	abscissa of the Cartesian plane
$y$	ordinate of the Cartesian plane
$z$	coordinates on the complex plane

## Greek Letters

$\beta$	mean distance among vortex-particles
$\Gamma$	circulation of a vortex, body or wake
$\delta$	desingularization parameter
$\epsilon$	relative error
$\lambda$	wavelength
$\mu$	dynamic viscosity
$\nu$	kinematic viscosity
$\rho$	density
$\sigma$	viscous core
$\phi$	irrotational field
$\psi$	solenoidal field
$\omega$	vector vorticity
$\omega$	bidimensional vorticity

## Superindices

$x^*$	non dimensional properties
-------	----------------------------

## Subindices

$i$	dummy index
$j$	dummy index
$k$	dummy index
$r$	radial velocity
$s$	source
$o$	observer
$n$	time-marching scheme step
$\mathcal{C}$	children box
$\infty$	free-stream property
$\theta$	tangential velocity

## Abbreviations

Re	Reynolds number
DVM	Discrete Vortex Method
FMM	Fast Multipole Method
RK	Runge-Kutta time-marching scheme
P2M	Particle-to-multipole
M2M	Multipole-to-multipole
M2L	Multipole-to-local
L2L	Local-to-local
L2P	Local-to-particle
P2P	Particle-to-particle
FF	Far-field
NF	Near-field
DP	Double precision (8 bytes)
QP	Quadruple precision (16 bytes)
RMS	Root-mean-square

## Other notations

$\mathcal{A}$	parameter of the FMM cost
$\mathcal{L}$	local expansion
$\mathcal{M}$	multipole expansion
$\mathcal{P}$	periodic cluster
$Re$	Real part of a number
$Im$	Imaginary part of a number

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>20</b>
1.1	Motivation and objectives . . . . .	21
1.1.1	Discrete vortex method . . . . .	23
1.1.2	Fast summation algorithms . . . . .	26
1.2	Contributions of the present work . . . . .	28
<b>2</b>	<b>DISCRETE VORTEX METHOD</b>	<b>29</b>
2.1	Fundamental equations . . . . .	30
2.2	Convection step . . . . .	33
2.2.1	Calculation of velocity component due to free-stream . . . . .	33
2.2.2	Calculation of velocity component due to solid boundaries . . . . .	34
2.2.3	Calculation of velocity component due to vortex wake . . . . .	36
2.2.4	Desingularization of potential vortex . . . . .	37
2.2.5	Time marching method . . . . .	40
2.2.6	Stability in time marching method . . . . .	42
2.3	Viscous diffusion step . . . . .	45
2.3.1	Diffusion models . . . . .	45
2.3.2	Vorticity generation . . . . .	47
2.3.3	Vorticity destruction . . . . .	48
<b>3</b>	<b>FAST MULTIPOLE METHOD</b>	<b>50</b>
3.1	Multi-level FMM . . . . .	54
3.1.1	Preprocessing . . . . .	54
3.1.2	Global refinement . . . . .	56
3.1.3	Computational cost . . . . .	61
3.2	Assessment of the FMM-DVM . . . . .	63
3.2.1	Preprocessing steps . . . . .	63
3.2.2	Desingularization models . . . . .	66
3.2.3	Example - flow around a circular cylinder . . . . .	68
3.3	Mathematical formulation . . . . .	72
3.3.1	Far-field . . . . .	73
3.3.2	Near-field induced velocity . . . . .	77
3.3.3	Total velocity . . . . .	77

<b>4</b>	<b>TEMPORAL EVOLUTION OF AIRCRAFT WAKE</b>	<b>78</b>
4.1	Initial condition . . . . .	80
4.2	Results . . . . .	82
4.2.1	Vortex model . . . . .	82
4.2.2	Discretization . . . . .	83
4.2.3	Fast Multipole Method parameters . . . . .	84
4.3	Vortex sheet roll-up . . . . .	86
4.3.1	Biot-Savart computations . . . . .	86
4.3.2	Numerical accuracy of the FMM . . . . .	88
4.3.3	Fast Multipole Method computations . . . . .	89
4.3.4	Computational time . . . . .	94
<b>5</b>	<b>PERIODIC SHEAR LAYER</b>	<b>97</b>
5.1	Numerical formulation . . . . .	97
5.1.1	Initial conditions and discretization . . . . .	98
5.1.2	Induced velocity by a vortex array . . . . .	99
5.1.3	Periodic FMM . . . . .	100
5.1.4	Desingularization . . . . .	106
5.2	Results . . . . .	107
5.2.1	Velocity profile . . . . .	107
5.2.2	Biot-Savart solution . . . . .	108
5.2.3	Periodic FMM's convergence . . . . .	114
5.2.4	Computational Time . . . . .	122
<b>6</b>	<b>FAST COTANGENT FUNCTION SUMMATION</b>	<b>124</b>
6.1	Exponential series expansion . . . . .	124
6.2	Power series expansion . . . . .	126
6.3	Fast summation algorithm . . . . .	128
6.3.1	Fast summation of the exponential series . . . . .	129
6.3.2	Fast summation of the power series . . . . .	131
6.3.3	Refinement of the initial domain . . . . .	135
6.4	Results . . . . .	139
6.4.1	Evaluation of velocity field . . . . .	139
6.4.2	Error analysis . . . . .	141
6.4.3	Computational time . . . . .	142
<b>7</b>	<b>CONCLUSION</b>	<b>146</b>
7.1	Suggestions for future work . . . . .	148

<b>References</b>	<b>149</b>
<b>APPENDIX A - FMM implementation</b>	<b>161</b>
A.1 Input . . . . .	161
A.2 Pre-processing . . . . .	161
A.3 Mapping . . . . .	165
A.4 Upward pass . . . . .	166
A.5 Downward pass . . . . .	167
A.6 Local-to-particle . . . . .	169
A.7 Particle-to-particle . . . . .	169
<b>APPENDIX B - Periodic FMM</b>	<b>170</b>
B.1 Multipole-to-multipole . . . . .	170
B.2 Multipole-to-local . . . . .	171
B.3 Local-to-local . . . . .	171

# 1 INTRODUCTION

Computational fluid dynamics, CFD, is the branch of fluid mechanics which solves problems involving fluid flows using numerical methods implemented in a computer. One should select the governing equations which describe the physics of interest and discretize them in an appropriate format to be solved using a numerical algorithm. Several models can be used depending if the flow to be analyzed is continuous or rarefied, compressible or incompressible, steady or unsteady, viscous or inviscid, etc. In order to determine the physical characteristics of a particular flow, one should analyze the relevant properties of this flow. In general, non-dimensional parameters are used to characterize a fluid flow. For example, the Knudsen number provides a ratio between the molecular mean free path and the characteristic length of the flow investigated, and it is considerably smaller than unity in a flow modeled as continuous. The Mach number indicates flow compressibility and it is given by the ratio between the local flow speed to the local speed of sound. If this number is considerably smaller than unit, the fluid can be considered incompressible. Another important non-dimensional parameter, which characterizes viscous flows, is the Reynolds number. It relates the effects of inertia to those of viscous diffusion. Usually, high Reynolds numbers are related to the presence of turbulence.

In the present work, we consider the solution of continuous, incompressible, unsteady viscous flows which represent an important class of practical problems in engineering. The incompressible form of the Navier-Stokes equations are chosen to accurately model the problems of interest. Three main classes of numerical methodologies are often found in literature for solving the Navier-Stokes equations. One is called direct numerical simulation, DNS, and it solves the Navier-Stokes equations without models or simplifications. In this methodology, all spatial and temporal scales of motion are resolved by the numerical method. If DNS is employed to solve turbulent flows, it needs to resolve from the largest, most energetic, scales of the flow, characterized by the boundary conditions from the geometry of interest, all the way to the Kolmogorov scales, which are the smallest flow structures and are the responsible for viscous dissipation. In order to apply DNS, one needs to discretize the flow governing equations using a fine computational mesh able to capture all the scales in the flow. Moreover, non-dissipative non-dispersive numerical methods need to be applied to minimize numerical dissipation. Although DNS can be used for fundamental studies of turbulence and its physics, it is also an expensive methodology which increases in cost with the Reynolds number since finer meshes are required to capture finer scales of the flow.

A second methodology solves the Reynolds-averaged Navier-Stokes equations, RANS, where a temporal filter is employed and the flow is solved for time-averaged properties. All the scales of turbulence are modeled to close the RANS equations which have additional terms compared to the Navier-Stokes equations. This methodology provides accessible computational costs when compared to DNS, and it can be employed in the solution of practical problems of engineering which usually have high Reynolds numbers.

Finally, a third methodology is large eddy simulation, LES, which resolves the largest scales of the flow and models the smallest scales. In LES, a spatial filter is applied to the Navier-Stokes equations and a set of new equations is solved. This methodology is more accurate than RANS since it resolves the most energetic scales which are anisotropic and model the smallest scales, which are more universal and are responsible for the diffusion processes in the flow. Although LES can be expensive if wall resolution is achieved, it still provides better computational costs than DNS.

Either of the methodologies described above can be solved using two different approaches, Eulerian and Lagrangian. In the Eulerian approach, the physical domain where the flow is to be solved for is kept fixed and a computational mesh discretizes the domain. The fluid flows through the boundaries of the domain which define the control volume. In general, the regions of interest are defined by mesh points (or volumes) which conform to the geometry. This approach is the most used in CFD. In the Lagrangian approach, the fluid elements are tracked as they flow and the control volumes are constantly changing. Both approaches have benefits and drawbacks. Eulerian methods require computational grids which can be cumbersome to generate for complex three-dimensional geometries. Furthermore, meshes with poor quality can compromise the flow solution through excessive numerical dissipation. However, these methods may allow easier reconstruction of numerical schemes with arbitrary order of accuracy and are more consolidated in literature. Lagrangian methods on the other hand may not require mesh generation. However, these methods may become expensive for realistic flow simulations and several aspects of Lagrangian methods still need to be investigated since they are not as consolidated as Eulerian methods.

## 1.1 Motivation and objectives

Several flows of practical interest in engineering include complex physics. Among these flows, one can cite those involving combustion, vortex-induced vibration, acoustics and turbulence. The solution of such flows is only possible through application of numerical methods which are time accurate since these problems are naturally unsteady with several frequency scales. Furthermore, since a broad range of spatial scales is also intrinsic of these

flows, one needs numerical algorithms with controlled dissipation. If Eulerian methods are employed to solve the aforementioned flows, fine computational grids and high-order numerical schemes are required to capture all dynamic scales of the problem. Moreover, small time steps are required to keep the simulation stable and time accurate. The computational cost of these unsteady simulations may be expensive since they generally need to run in high-performance parallel computers for considerable amount of time.

Lagrangian methods, such as the discrete vortex method, DVM, do not require the use of a computational mesh. In the DVM, the Navier-Stokes equations are solved in the vorticity form and individual vortex particles are transported with the flow. These vortex particles are generated along solid boundaries where vorticity is present. They also interact with each other inducing flow velocities which are governed by the Biot-Savart law. The convection and diffusion processes can be solved separately in the DVM and the calculation of non-linear terms in the Navier-Stokes equation can be avoided through application of the material derivative, inherent of Lagrangian methods. Several aspects of the DVM formulation are still under debate in literature, such as those involving vortex generation, destruction and diffusion. The computational cost of the DVM is mainly related to the convection process in the Navier-Stokes equation since a cloud with  $N$  vortex particles has  $N^2$  interactions (each one interacts with all other particles). Since more accurate solutions usually require clouds with several vortex particles, the method can become impractical for simulations of realistic flows in complex geometries.

In the present work, we apply a Lagrangian method to solve canonical flows with practical application in engineering problems. The discrete vortex method is employed in the simulations of a Trefftz plane and a time evolving shear layer. The Trefftz plane represents the formation of wing tip vortices which appear in the wake of an airplane and the time evolving shear layer occurs in problems of combustion and hydrodynamic instability. A numerical tool which solves a two-dimensional, incompressible, viscous flow is developed. The use of a Lagrangian approach with the DVM avoids computational mesh generation and further problems associated with numerical dissipation. However, as already mentioned, the DVM is expensive due to its  $O(N^2)$  computational cost.

One of the objectives of this work is to develop a fast Lagrangian numerical tool. Here, a fast summation algorithm is implemented together with the DVM in order to reduce its computational cost. The fast multipole method, FMM, is the method of choice and it is implemented together with the Navier-Stokes equations in vorticity form. The FMM has been applied in the literature to accelerate the solution of several problems involving potential flows, acoustics, electromagnetism and heat transfer. In general, these problems are solved for steady state solutions. When the FMM is employed together with the DVM, the method needs to be implemented in a time accurate frame since individual vortex

particles move according to the local flow velocity. Another objective of this work is to investigate the effects of motion and interaction of vortex particles using the combination of DVM and FMM. In order to couple these two algorithms, several modifications are proposed in the implementation of the original FMM. A final objective of this work is to investigate the sources of errors from the algorithms implemented including the DVM and the FMM. A quick overview of both the discrete vortex method and the fast multipole method is presented below, with a literature survey.

### 1.1.1 Discrete vortex method

The discrete vortex method is based on the vorticity transport equation, where local vorticity is solved for. The method employs a discretization of vorticity using  $N$  vortex particles with individual constant circulation. For example, in a solid boundary there is the creation of vortex elements concentrated in the boundary layer due to the no-slip condition. The vortex elements are then convected with the local flow velocity along the wake of the solid boundary. As vortex dynamics obeys the Helmholtz theorem, it is necessary to track all the vortex elements. To do so, the velocity field is evaluated by the contribution of free-stream, solid bodies and vortex-vortex interactions. For the last contribution, one must solve the Biot-Savart law using a convolution step with computational cost proportional to  $O(N^2)$ .

The DVM has several applications which extend from unbounded problems such as a free jet to more complex cases with multiple solid bodies. The method was initially applied to problems involving continuous vorticity layers which were discretized by  $N$  potential vortex particles of inviscid flows (Rosenhead, 1932; Westwater, 1936). For a small  $N$ , the particles showed a nice and smooth path. However, when the number was increased, the distance between elements was reduced and singular effects from the non-physical potential vortex led to their chaotic motion (Birkhoff and Fisher, 1959; Birkhoff, 1962; Takami, 1964; Moore, 1971). Hence, with more elements the solution was worse and some authors had a disbelief in the method.

Years later, smoother vortex models with localized viscosity in their core were investigated, leading to more accurate results due to the smoother velocity field. The main models used are the Lamb-Oseen vortex with localized diffusion and the vortex blob based on Hilbert transform with conjugate Poisson kernel (Chorin, 1973; Krasny, 1986a; Batchelor, 2000; Abid and Verga, 2002; Holm *et. al.*, 2006; Baker and Pham, 2006).

Several authors studied the effects of convergence and other numerical errors in the discrete vortex method. For instance, Sethian (1988) shows the flow in a backward facing step, where the increase in Reynolds number generates several structures which

are advected with the flow, in good agreement with experiments. Other studies are covered in the work of Hald and Del Prete (1978); Leonard (1980); Beale and Majda (1985); Anderson and Greengard (1985); Sethian and Ghoniem (1988); Sparlat (1988); Hou *et. al* (1991); Shintani and Akamatsu (1994); Barba *et. al* (2003).

Extensions of the discrete vortex method for three dimensions were proposed (Anderson and Greengard, 1985; Leonard, 1985) including applications of flows past spheres (Nakanishi and Kamemoto, 1993; Ploumhans *et. al*, 2002), aircraft wakes (Chatelain *et. al*, 2008) and free turbulence (Yokota and Obi, 2010). Formulations of the DVM to solve compressible flows were proposed and implemented to study aeroacoustics problems (Eldredge *et. al*, 2001; Eldredge, 2002). The DVM was also employed in the simulation of flows involving heat transfer, where the energy equation was also solved with the method (Ghoniem *et. al*, 1988; Ogami, 2001; Malakhova *et. al*, 2010).

Inviscid vorticity layers were studied to represent models of aircraft wakes. In this case, the three dimensional evolution of the wake is simplified to an initial value problem in two dimensions as shown by Smith (1986). This problem is called Trefftz plane and it is also investigated in the present work. The solution is obtained by replacing a continuous vorticity layer by discrete vortex particles (Chorin and Bernard, 1973; Fink and Soh, 1978; Krasny, 1987). An overview of this problem is also presented by Drela (2014).

Other inviscid problems were also solved using the DVM. For example, infinite vorticity layers were investigated considering non-miscible fluids with different properties. The individual layers were placed in contact and, since no dissipation mechanism was present, small perturbations amplified forming Kelvin-Helmholtz instabilities (Krasny, 1986b,a; Ghoniem *et. al*, 1988; Tryggvason, 1989; Tryggvason *et. al*, 1991; Shelley, 1992; Abid and Verga, 2002). This type of phenomenon occurs in several flows of practical interest such as those involving combustion (Herrmann, 2005) and it is also investigated in the present work. Other applications of the DVM includes vortex formation in edges of tubes (Nitsche and Krasny, 1994), and two-dimensional jets (Shmagunov, 2012).

The discrete vortex method was also applied to solve viscous flows including solid surfaces such as cylinders and flat plates. In these cases, the walls worked as vorticity generators due to the enforcement of no-slip boundary condition. However, the creation of vorticity is not a consensus among authors and several issues may appear in this process such as the treatment of vortex particles crossing a solid boundary. For instance, some authors consider the vortex to be rigid bodies which are reflected if they hit the body, while other eliminate those inside the body and their circulation is distributed among the new-born. For details on vorticity generation and destruction, one should see the work of Chorin (1978); Porthouse (1983); Koumoutsakos *et. al* (1994); Shiels (1998); Kamemoto (2004); Ramachandran *et. al* (2007); Barba and Cooper (2009).

Another aspect of the DVM includes the influence of solid boundaries on the velocity field. This effect is accounted in several ways and, in general, it is overcome by solving the boundary element method, BEM (Martensen, 1959; Hess and Smith, 1962; Hess, 1972; Drela, 2014), which can handle flows around arbitrary geometries. Other methods which can account for the influence of solid boundaries include conformal mappings (Kuwahara, 1973; Sarpkaya, 1975; Chein and Chung, 1988), the method of images (Gerrard, 1967), or other numerical methods such as those involving radial basis functions (Barba and Cooper, 2009). However, some of these methods have limited applications and can be only used for simple geometries. Finally, simplified methods involving vorticity generation are also proposed in literature and they consider the creation of vortex particles only at detachment points of blunt bodies (Gerrard, 1967; Wong, 1985; Dewey *et. al*, 2014). In order to apply such methods, one needs to previously know some characteristics of the flow investigated.

Examples of applications of the DVM for flow simulations around cylinders are discussed by Chorin (1973); Sarpkaya and Schoaff (1979); Koumoutsakos (1993); Tsutsui *et. al* (1997); Ploumhans and Winckelmans (2000). Bodies in arbitrary movement have been studied using the DVM by Yokoi and Kamemoto (1994); Guvernyuk and Dynnikova (2007); Malakhova *et. al* (2010); Dynnikov *et. al* (2014). Other authors employed the DVM to investigate the flow past multi-element airfoils (Golia and Viviani, 2011), flat plates parallel or perpendicular to the flow (Kuwahara, 1973; Sarpkaya, 1975; Kiya and Arie, 1977; Chorin, 1978; Chein and Chung, 1988; Walther and Larsen, 1997; Andronov *et. al*, 2007) and airfoils in flapping flight (Eldredge, 2005; Dewey *et. al*, 2014).

Viscous problems were also investigated in the absence of walls such as in mixing layers with spatial evolution (Ashurst, 1977). In viscous problems, the diffusion step may be solved using several techniques. The first method proposed is the Random Walk Method, a stochastic model where the particles have a displacement in any direction based on Reynolds number. Other scheme increases the viscous core of the vortex and their subsequent division when they become too large. Also, the circulation may be passed from one vortex for those in the surrounding. However, while some diffusion methods are meshless and, therefore, coherent with the DVM essence, other methods require computational grids which may add artificial diffusion. For more details on the most important methodologies, one should see the work of Greengard (1985); Ghoniem (1985); Bernard (1995); Rossi (1996); Shankar and Dommelen (1996); Kida and Nakajima (1998); Shiels (1998); Ploumhans and Winckelmans (2000); Barba (2004).

Further details about the discrete vortex method can be found in Leonard (1980); Aref (1983); Sparlat (1988); Lewis (1991); Saffman (1992); Stock (2007).

### 1.1.2 Fast summation algorithms

The discrete vortex method requires the calculation of several convolutions in order to account for the interaction among individual vortex particles. These calculations represent the bottleneck of the method in terms of computational cost since they grow with the square of the number of particles in the cloud. Therefore, alternative methods which employ fast summation algorithms are required to reduce the computational cost of the DVM. In this sense, several fast algorithms were originally proposed to solve problems of gravitational interactions in celestial mechanics (Appel, 1985; Barnes and Hut, 1986; Dehnen, 2002). These methods could be modified to solve the DVM, however, in these algorithms, high accuracy was not an important aspect. For vortex dynamics, high accuracy in the calculation of convolutions is a point of paramount importance since small errors can be amplified and modify the flow dynamic.

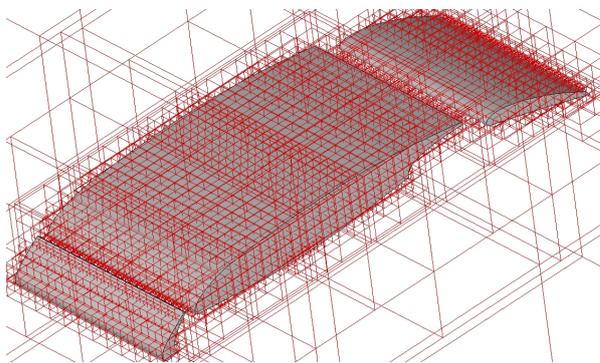
The vortex-in-cell method (Christiansen, 1973) is a more accurate algorithm and it can be modified for the solution of the DVM. It requires the use of a computational mesh and each vortex is replaced by a contribution at the four surrounding nodes of this mesh. A fast Poisson solver is used to calculate the stream function at the mesh points and, then, interpolation is used to calculate the velocity and displacement for each vortex. This method adds excessive numerical diffusion to the simulation and it was employed by few authors in literature (Milinazzo and Saffman, 1977; Tryggvason, 1989).

The Pseudoparticle method (Anderson, 1992) is a fast algorithm which seeks a better accuracy without requiring the use of a computational mesh. However, it has limited applications and cannot overcome the fast multipole method in accuracy and computational cost.

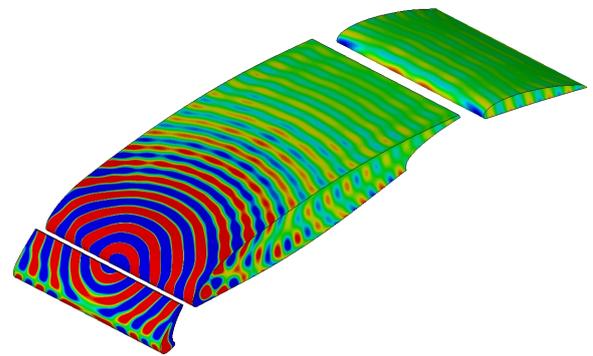
The FMM was proposed by Greengard and Rokhlin (1987); Carrier *et. al* (1988) and is listed as a top 10 algorithm of the 20th century (Cipra, 2000). It reduces the cost of convolutions from  $O(N^2)$  to  $O(N)$  using divide-and-conquer strategies. Instead of using particle-to-particle operations, the method performs computations among groups of particles. With the FMM, unsolvable problems of computational cost  $O(N^2)$  became viable due to cost reduction to  $O(N)$ . The method found applications in several problems of physics and engineering (potential flow, acoustics, electromagnetism, fracture mechanics, Stokes flow). For example, the method was extended for the solution of the boundary element method to solve problems of potential flows and acoustic scattering around arbitrary bodies or to solve the Ffowcs Williams and Hawkings equation in problems of acoustic propagation (Wolf, 2011). The FMM was also applied together with the method of moments for the calculation of electric charges and electromagnetic scattering (Coifman *et. al*, 1993). Problems with periodic boundary conditions can be solved using

the FMM, where the region of interest is replicated in  $n$ -dimensions according to the problem modeling. Investigations on this implementations are covered in the work of Lambert *et. al* (1996); Kudin and Scuseria (1998); Rodin and Overfelt (2004); Kurzak and Pettitt (2006); Kabadshow (2010); Yokota and Obi (2010).

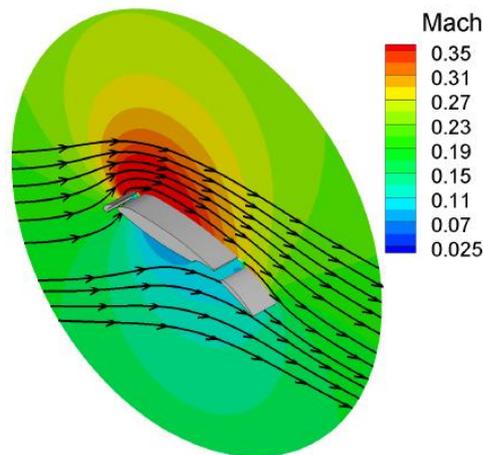
The idea of FMM to compute the interactions among groups of elements is shown through the adaptive refinement of several boxes around a multi-element airfoil, 1.1(a). These boxes contain the clusters of boundary elements which interact with each other. In Fig. 1.1(b) and (c), respectively, one can see acoustic scattering and potential flow around the airfoil.



(a) FMM boxes around a multi-element airfoil.



(b) Acoustic scattering around a multi-element airfoil.



(c) Potential flow around a multi-element airfoil.

Figure 1.1: Applications of the fast multipole method - extracted from Wolf (2011).

More details on the FMM and its applications can be found in Greengard and Rokhlin (1987); Carrier *et. al* (1988); Cheng *et. al* (1999); Darve (2000); Nishimura (2002); Ogami (2002); Gumerov and Duraiswami (2005); Wolf (2011).

## 1.2 Contributions of the present work

One of the main contributions of this work is the development of a numerical tool which couples the fast multipole method to the discrete vortex method. This coupling reduces the computational cost of the convolutions from the Biot Savart law inherent from the DVM. A detailed study of the accuracy of the numerical algorithm is provided including an analysis of the effects of time marching schemes on the error of the DVM. We also provide a complete analysis of the parameters which control the errors of the fast multipole method. The choice of these parameters is investigated to obtain machine precision for dynamic problems. This is a new contribution of the present work.

Another contribution of this work includes a study on aspects of desingularization of the potential vortex in the discrete vortex method. This study is important for the coupling between the DVM and the FMM and it is also a new contribution of the current work.

Periodic problems of infinite shear layers are solved in this work with a novel FMM implementation which reduces the error in truncation of periodical boundary conditions. This is achieved by a fine addition of the contribution of symmetric boundary conditions, which avoids a spurious precession of the infinite shear layer.

Lastly, an alternative scheme to solve the cotangent function, which is the analytical solution of the vortex replication that satisfies the boundary conditions in periodic problems, is investigated. This approach is compared to the crude replication of particles, indicating that better results can be achieved in this alternative method.

## 2 DISCRETE VORTEX METHOD

The discrete vortex method, DVM, is a numerical method which solves the vorticity transport equation for fluid flow simulations. In this equation, the Navier-Stokes equations are written in the vorticity form and, therefore, the pressure term in these equations vanish. The method employs a Lagrangian approach in order to solve the vorticity equation for discrete point vortices which are used to represent the vorticity field. Hence, the DVM is a meshless method which solves the fluid flow by tracking the individual vortices, according to the Helmholtz theorem. To do so, the velocity field must be evaluated by the contribution of the free-stream, solid boundaries and vortex-vortex interactions using the Biot-Savart law. For a compact distribution of vorticity, limited to a specific region, there is no need to solve the entire flow domain; only the region where vorticity is present is solved, *i.e.*, at each vortex location.

The most common numerical methods employed in computational fluid dynamics, CFD, require grids (structured or unstructured) where derivatives or flux integrals are carried out. Since the early stages of CFD, mesh generation has been an important step of the whole numerical simulation. For realistic engineering problems, it can be an expensive step since complex geometries may require accurate representation of specific portions. For example, in simulations of compressible turbulent flows, one needs to refine the computational mesh in regions where boundary layers and wakes are present. Furthermore, it is important to refine the mesh along regions with shock waves and contact surfaces. In these regions, large gradients of properties need to be accurately captured by the mesh and the numerical algorithm. Usual methods of CFD have intrinsic properties regarding artificial dissipation which may negatively affect the simulation of problems with a broad range of scales, such as turbulence, acoustics and compressible flows.

In this context, the absence of a computational grid in Lagrangian methods allows the simulation of arbitrary bodies without the cost of mesh generation. However, these methods also present drawbacks. In the DVM, for example, the memory cost to keep track of all  $N$  discrete vortex particles is proportional to  $O(N)$ , but the computational time is proportional to  $O(N^2)$  since a convolution step is required to compute the velocity field using the Biot-Savart law. This is a major disadvantage of this method. In this chapter, the mathematical formulation of the discrete vortex method is presented, following the deduction from Saffman (1992) and Kundu *et. al* (2012). Also, an overview of the convection and diffusion steps in the DVM is discussed along with comments on the numerical stability of the method.

## 2.1 Fundamental equations

The flow of an incompressible, Newtonian fluid with constant properties, is governed by the continuity equation through mass conservation, Eq. 2.1, and the Navier-Stokes equations, Eq. 2.2. One should mention that these equations are presented here in their incompressible form.

$$\nabla \cdot \mathbf{u} = 0 , \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \frac{1}{\rho} \left( -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \right) . \quad (2.2)$$

Here,  $\mathbf{u}$  represents the velocity vector,  $\mu$  is the dynamic viscosity coefficient,  $p$  is the hydrodynamic pressure and  $\mathbf{g}$  represents gravity acceleration.

According to Helmholtz decomposition, a vector field  $\mathbf{u}(\mathbf{x}, t)$  can be decomposed as the sum of an irrotational scalar field  $\phi$  and a solenoidal vector field  $\boldsymbol{\psi}$ :

$$\mathbf{u}(\mathbf{x}, t) = \nabla \phi + \nabla \times \boldsymbol{\psi} . \quad (2.3)$$

The irrotational term  $\phi$  represents the potential flow, which is the inviscid region of the flow in regions outside wakes and boundary layers. For a solenoidal flow, the following equation is satisfied:

$$\nabla \times (\nabla \phi) = 0 , \quad (2.4)$$

where, by definition, a solenoidal vector field is divergence free:

$$\nabla \cdot \boldsymbol{\psi} = 0 . \quad (2.5)$$

The curl of Eq. 2.3:

$$\nabla \times \mathbf{u}(\mathbf{x}, t) = \nabla \times (\nabla \phi) + \nabla \times (\nabla \times \boldsymbol{\psi}) , \quad (2.6)$$

where the first term can be simplified using Eq. 2.4 and the second one is expanded according to vector calculus identities:

$$\nabla \times \mathbf{u}(\mathbf{x}, t) = \nabla (\nabla \cdot \boldsymbol{\psi}) - \nabla^2 \boldsymbol{\psi} , \quad (2.7)$$

and, from Eq. 2.5, it is possible to obtain

$$\nabla \times \mathbf{u}(\mathbf{x}, t) = -\nabla^2 \boldsymbol{\psi} , \quad (2.8)$$

where the vorticity is defined as

$$\nabla \times \mathbf{u} = \boldsymbol{\omega} , \quad (2.9)$$

leading to

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega} . \quad (2.10)$$

Furthermore, from the curl of Eq. 2.9, vorticity and velocity are related such that:

$$\nabla \times \boldsymbol{\omega} = \nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u} , \quad (2.11)$$

and, for an incompressible flow, Eq. 2.1 leads to the following Poisson equation:

$$\nabla \times \boldsymbol{\omega} = -\nabla^2 \mathbf{u} . \quad (2.12)$$

For a three-dimensional field, the above Poisson equation can be solved as:

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{4\pi} \int_{V'} \frac{\boldsymbol{\omega}(\mathbf{x}', t) \times (\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} dV' , \quad (2.13)$$

while, for a two-dimensional problem, its solution is given by:

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{2\pi} \int_{S'} \frac{\boldsymbol{\omega}(\mathbf{x}', t) \times (\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^2} dS' , \quad (2.14)$$

where  $S'$  and  $V'$  represent a two-dimensional surface and a three-dimensional volume, respectively, around the vorticity field. Thus, from the vorticity field  $\boldsymbol{\omega}(\mathbf{x}', t)$  is possible to solve the velocity field  $\mathbf{u}(\mathbf{x}, t)$  anywhere in the flow domain.

For incompressible flow, the curl of Eq. 2.2 leads to the vorticity equation, Eq. 2.15. The solution of this equation simplifies, in some aspects, that of Eq. 2.2 since it no longer requires the calculation of the pressure field  $\mathbf{p}$ .

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} . \quad (2.15)$$

For a two-dimensional problem, the vorticity becomes a scalar which is expressed using only the  $z$ -vorticity component, Eq. 2.16.

$$\boldsymbol{\omega} = (0, 0, \omega) = \omega \hat{\mathbf{k}} , \quad (2.16)$$

and, in this case, the velocity vector only has components along the  $x$  and  $y$ -axis

$$\mathbf{u} = (u, v, 0) . \quad (2.17)$$

Furthermore, the vortex stretching term,  $(\boldsymbol{\omega} \cdot \nabla)\mathbf{u}$ , is null in the two dimensional problem since the velocity normal to the  $x - y$  plane ( $z$  direction) is null. This can be shown by

$$(\boldsymbol{\omega} \cdot \nabla)\mathbf{u} = \omega(\hat{\mathbf{k}} \cdot \nabla)\mathbf{u} = \omega \left( \frac{\partial \mathbf{u}}{\partial z} \right) = 0 , \quad (2.18)$$

which leads to a simplified equation:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \nabla^2 \omega . \quad (2.19)$$

This equation can be nondimensionalized based on a characteristic length scale,  $x_c$ , velocity,  $u_c$ , and time,  $t_c$ , giving the respective non-dimensional variables,  $x^*$ ,  $u^*$  and  $t^*$ :

$$x^* = \frac{x}{x_c} , \quad (2.20)$$

$$u^* = \frac{u}{u_c} , \quad (2.21)$$

$$t^* = \frac{t_c u_c}{x_c} . \quad (2.22)$$

For an incompressible fluid, the vorticity transport equation in non-dimensional variables is finally given by:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \frac{1}{Re} \nabla^2 \omega , \quad (2.23)$$

where the non-dimensional superscripts  $*$  were dropped for a simplified notation.

Chorin (1973) proposed a numerical solution for viscous flows at high Reynolds numbers using the DVM, where his method is exemplified by the flow around a circular cylinder. In his algorithm, he solves separately the inviscid and viscous terms of Eq. 2.23 in two fractional steps.

1. The first step considers the flow to be inviscid, *i.e.*,  $Re \rightarrow \infty$ , leading to:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = 0 . \quad (2.24)$$

2. The second step solves the flow including only the viscous effects:

$$\frac{\partial \omega}{\partial t} = \frac{1}{Re} \nabla^2 \omega . \quad (2.25)$$

In this work, we will follow the ideas of Chorin (1973) and employ the DVM for the solution of the two-dimensional vorticity equation for several problems of interest. The separate steps of convection and diffusion will be discussed below in more details.

## 2.2 Convection step

The DVM is a meshless method which solves the vorticity equation in a Lagrangian context, tracking individual vortex particles. Therefore, we make use of the material derivative in Eq 2.24 and avoid the solution of the non-linear term in the equation since

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \frac{D\omega}{Dt} = 0 . \quad (2.26)$$

With the material derivative in Eq. 2.26, it is natural to use a Lagrangian approach. This way, discrete vortex particles are used to mimic the vorticity field, and their dynamics obey the Helmholtz theorem which says that vortex lines move with the local velocity field and their intensity remain constant in time (for inviscid flows only). So it is only necessary to know the position  $\mathbf{x}_i$  and circulation  $\Gamma_i$  of the  $N$  vortices, which leads to a memory cost proportional to  $O(N)$ , which is inexpensive.

To transport the vortex particles according to Helmholtz theorem, it is necessary to evaluate the local fluid velocity for all vortices. In the presence of solid boundaries or uniform incident flows, the potential scalar  $\phi$  in Eq. 2.3 is finite and contributes to the velocity field. Since the present equation is linear, it is possible to simply sum all the individual contributions of the vortex wake, solid body and uniform flow to obtain the total velocity field at any point in the flow domain as:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\infty + \mathbf{u}_{body} + \mathbf{u}_{wake} . \quad (2.27)$$

### 2.2.1 Calculation of velocity component due to free-stream

In the DVM, any incident parallel flow may be accounted for using the solution for the potential velocity

$$\phi = U_\infty x , \quad (2.28)$$

or, in terms of streamlines,

$$\psi = U_\infty y . \quad (2.29)$$

Furthermore, if the flow has an angle of attack given by  $\alpha$ , the flow components  $u$  and  $v$  can be obtained by

$$u = U_\infty \cos(\alpha) , \quad (2.30)$$

and

$$v = U_\infty \sin(\alpha) . \quad (2.31)$$

## 2.2.2 Calculation of velocity component due to solid boundaries

In potential flow problems, the calculation of velocity components due to solid boundaries can be performed by the Method of Images for simplified flow configurations such as that of a vortex near a horizontal wall or a cylindrical boundary (Gerrard, 1967). A conformal transformation can be used when the flow is in the presence of edges or other configurations such as in finite plates, polygons and ellipses (Chein and Chung, 1988; Kuwahara, 1973; Sarpkaya, 1975). A well known solution of conformal transformation is that from Joukowski airfoils, shown by Kundu *et. al* (2012).

The Boundary Element Method, BEM, can be used to solve the potential flow over complex geometries, with any number of bodies, through the solution of the Laplace equation, Eq. 2.4, in a boundary integral form. In order to solve the integral equations, the solid surfaces were discretized in panels with distributions of singularities with unknown intensities. Martensen (1959) proposed a version of the BEM in two dimensions to solve lifting surfaces by using vorticity layers distributed on the boundary elements. Latter, it was extended by Hess and Smith (1962) to study potential flow around arbitrary geometries with zero lift, in two or three dimensions, by a distribution of sources and sinks, which deflect the streamlines. The method increased its capabilities when Hess (1972) used doublets to solve for lifting surfaces as well. The kind of singularity used depends on the problem to be solved, and they can be placed at the center of the panel. Also, they can have a constant distribution (zeroth order) or smoother distributions such as linear, quadratic or even higher orders.

For a source distribution, it is more suitable to use a Neumann boundary condition for the potential function, where  $\mathbf{n}$  is the panel normal vector. The boundary condition is written as:

$$\frac{\partial \phi}{\partial \mathbf{n}} = U_{\infty} \cdot \mathbf{n} = U_{\mathbf{n}} \quad \text{at solid surface S,} \quad (2.32)$$

$$\frac{\partial \phi}{\partial \mathbf{x}} = U_{\infty} \quad \text{for } \mathbf{x} \rightarrow \infty . \quad (2.33)$$

For vorticity distributions, a streamline function with Dirichlet boundary condition can be applied as:

$$\psi \quad \text{constant at surface S,} \quad (2.34)$$

$$\frac{\partial \psi}{\partial \mathbf{x}} = U_{\infty} \quad \text{for } \mathbf{x} \rightarrow \infty . \quad (2.35)$$

After solving a linear system for the  $n$  singularities distributed along the  $n$  panels, the velocity field at any point can be calculated. Therefore, it is possible to account for the induced velocity by each panel at all vortex particles with a computational cost proportional to  $O(nN)$ . The results from Katz and Plotkin (1991) deductions are shown in this section using a panel local coordinate system which is oriented following a counter clockwise direction, Fig. 2.1:

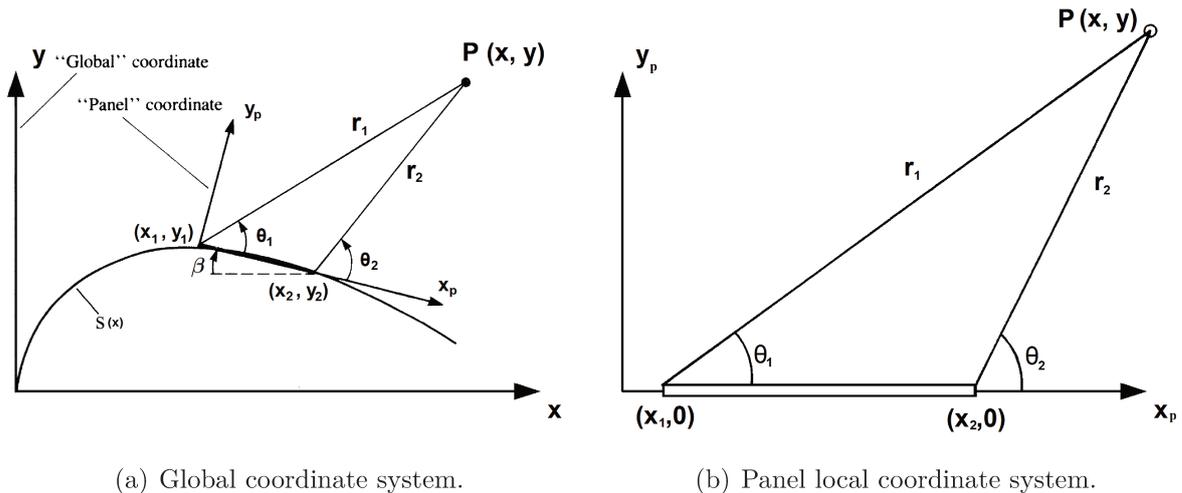


Figure 2.1: Coordinate system for BEM.

In order to rotate to the local from the global coordinate system, according to the panel slope  $\theta_i$ , as well as a translation based on the coordinates  $x_o$  and  $y_o$  of the panel origin in the global coordinate system, one must solve

$$\begin{Bmatrix} x_p \\ y_p \end{Bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{Bmatrix} x - x_o \\ y - y_o \end{Bmatrix}. \quad (2.36)$$

The velocity components  $u$  and  $v$  in the local, rotated coordinate system, indicated by the subindex  $p$ , induced by a constant distribution of sources is given, respectively, by

$$u_p = \frac{\sigma_i}{2\pi} \log \left( \frac{r_1}{r_2} \right), \quad (2.37)$$

$$v_p = \frac{\sigma_i}{2\pi} (\theta_2 - \theta_1). \quad (2.38)$$

One must rotate back to the global coordinate system, leading to the velocity induced on each vortex particle by the distribution of sources on the panels, calculated as

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{bmatrix} \begin{Bmatrix} u_p \\ v_p \end{Bmatrix}. \quad (2.39)$$

### 2.2.3 Calculation of velocity component due to vortex wake

The velocity induced at a point  $(x_i, y_i)$  by a potential vortex at  $(x_j, y_j)$ , with circulation defined by  $\Gamma > 0$  in clockwise sense, is given by the Biot-Savart law and it is purely tangential, such that there is no radial component.

$$V_\theta = -\frac{\Gamma_j}{2\pi r_{ij}} \quad (2.40)$$

$$V_r = 0 \quad (2.41)$$

When the distance  $r_{ij}$  between points  $i$  and  $j$  tends to 0, one should observe a singularity

$$r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2. \quad (2.42)$$

Also, using complex notation, the distance between two points  $i$  and  $j$  is

$$z_{ij} = (x_i - x_j) + i(y_i - y_j). \quad (2.43)$$

The velocity field is given by

$$w_i = i \frac{\Gamma_j}{2\pi z_{ij}}, \quad (2.44)$$

which can be further manipulated using the complex conjugated  $\bar{z}_{ij}$  of  $z_{ij}$ . Hence, one can obtain the  $u$  and  $v$  components by both real and imaginary parts, respectively, of

$$w_i = u_i - iv_i = \frac{\Gamma_j}{2\pi r_{ij}^2} [i(x_i - x_j) + (y_i - y_j)]. \quad (2.45)$$

The black arrows in Fig. 2.2 illustrate the velocity field induced by the vortex  $j$ .

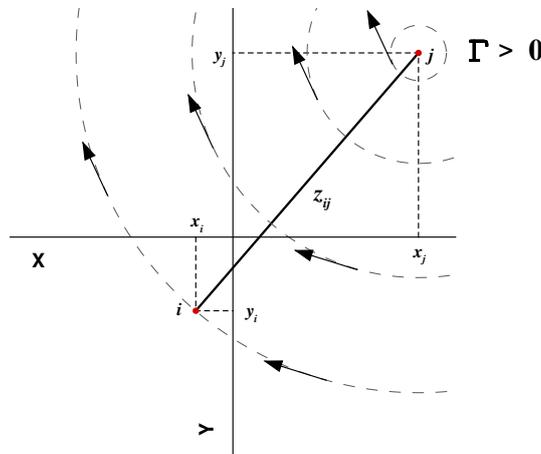


Figure 2.2: Representation of vortex interaction in Argand diagram for complex numbers.

As each vortex induces a velocity in all the other vortices, the interactions among all  $N$  vortex elements lead to the expensive computational simulation cost of  $O(N^2)$ . This is represented by the following equation which shows the calculation of the velocity field  $\mathbf{u}$  by a convolution in all particles:

$$\mathbf{u} = \sum_{i=1}^N \sum_{j=1}^N \mathbf{i} \frac{\Gamma_j}{2\pi z_{ij}} . \quad (2.46)$$

Leonard (1980) shows that the numerical error of the DVM is proportional to  $O(\beta^2)$ , where  $\beta$  is the average spacing between two vortices. A more accurate discretization of the vorticity field can be obtained if  $\beta \rightarrow 0$  and  $N \rightarrow \infty$ . However, since the computational cost of the method is proportional to  $O(N^2)$ , there is a strong disadvantage compared to usual CFD methods which employ finite stencils and computational grids.

This is where the fast multipole method, FMM, is useful to reduce the computational cost of the Biot-Savart operations to  $O(N)$ . Instead of using particle-to-particle operations, the FMM performs computations among groups of particles. More details on the method are presented in Chapter 3.

## 2.2.4 Desingularization of potential vortex

To overcome the singularity of a potential vortex, Eq. 2.44, a smooth model for the vorticity distribution  $\omega$  should be implemented. The singular distribution of vorticity lead to the disbelief in results produced by the DVM in the past, but the application of a smooth kernel showed that the method can provide accurate physical results. Earlier investigations using potential vortices are covered in the work of Rosenhead (1932); Westwater (1936); Takami (1964); Moore (1971); Chorin (1973); Birkhoff (1962).

One of the most used models with non-singular velocity field is the Lamb-Oseen vortex, which has a second-order Gaussian core  $\sigma$  to represent the core diffusion, based on the solution of vorticity diffusion equation for a radial distance  $r$ . If the rotational region  $\omega(r)$  does not change in time, *i.e.*, it is not allowed to further diffuse, the viscous core  $\sigma$  is fixed. Hence, one can write

$$\omega(r) = \frac{\Gamma_j}{\pi\sigma^2} \exp\left(-\frac{r^2}{\sigma^2}\right) . \quad (2.47)$$

The tangential velocity profile based on the radial distance  $r$  is then given by

$$V_\theta = -\frac{\Gamma}{2\pi r} \left[1 - \exp\left(-\frac{r^2}{\sigma^2}\right)\right] . \quad (2.48)$$

Also, it is possible to define a non-dimensional viscous core,  $c^*$ , calculated as

$$c^* = \frac{r}{\sigma} . \quad (2.49)$$

Figure 2.3(a) shows that smaller cores,  $\sigma \rightarrow 0$ , tend to recover the potential vortex pattern, which is a Dirac delta vorticity distribution. Figure 2.3(b) shows a self-similar profile of vorticity distribution of a Lamb-Oseen vortex, indicating that no matter how diffused is the core, it keeps a Gaussian shape based on the dimensionless viscous core  $c^*$ .

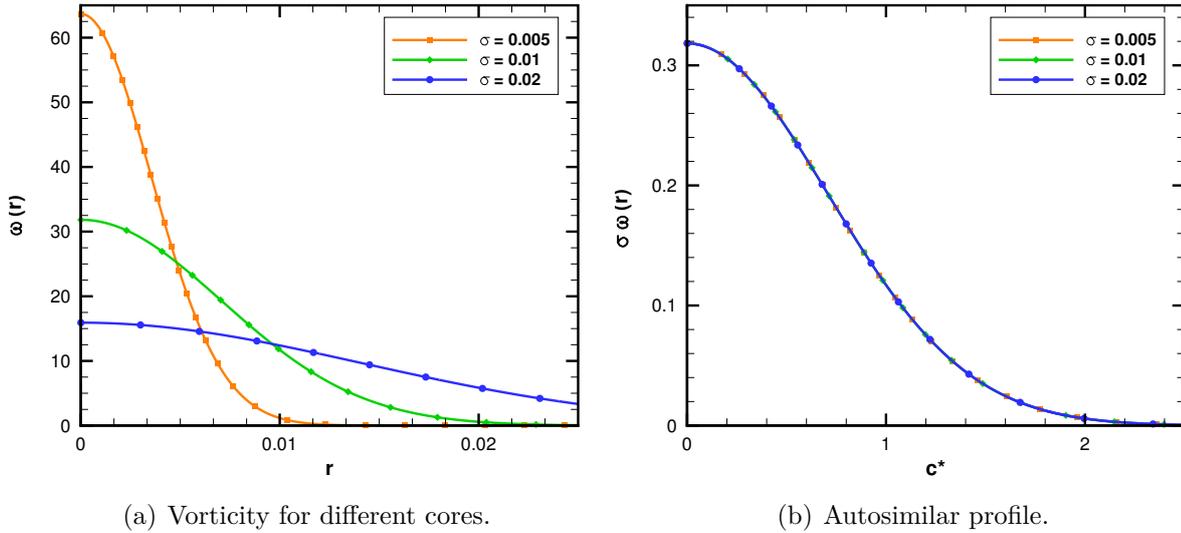


Figure 2.3: Vorticity distribution in a Lamb-Oseen Vortex.

For more details on this particular vortex model, one should see Batchelor (2000), Saffman (1992) or Kundu *et. al* (2012), and for details on higher-order kernels, see Beale and Majda (1985).

Another desingularization model is the vortex blob, given by Eq. 2.50. It was proposed by Krasny (1986b) and it is based in the Hilbert transform with conjugate Poisson kernel, where  $\delta$  is an arbitrary parameter. This model converges to the potential vortex when the desingularization term  $\delta$  tends to 0

$$V_\theta = -\frac{\Gamma}{2\pi r} \left( \frac{r^2}{r^2 + \delta^2} \right) = -\frac{\Gamma}{2\pi r} \left( \frac{1}{1 + \delta^2/r^2} \right) . \quad (2.50)$$

However, this non-singular model relies on a pure numerical artifact to smooth the velocity field induced by a point-vortex, as pointed by Krasny (1987). Its equivalent non-dimensional viscous core is given by:

$$c^* = \frac{r}{\delta} . \quad (2.51)$$

Other alternatives to smooth the vorticity distribution are shown by Chorin (1973), who truncates the velocity profile in regions very close to the vortex; by Holm *et. al* (2006) who proposed a vortex using an Euler-alpha regularization; or the application of a Rankine vortex, where the velocity varies linearly inside the viscous core. However, only a few works apply these models.

A further examination of both models is performed using Eqs. 2.40, 2.48 and 2.50. Since they are used to approach the potential vortex, their relative deviation  $\epsilon$  in induced velocity  $w$  of Lamb and blob vortex models from the potential one can be computed, respectively, as:

$$\epsilon_{Lamb} = \frac{w_{pot} - w_{Lamb}}{w_{pot}} \quad (2.52)$$

and

$$\epsilon_{blob} = \frac{w_{pot} - w_{blob}}{w_{pot}} . \quad (2.53)$$

This leads to Lamb-Oseen and vortex blob deviations to the potential vortex, given, respectively, by

$$\epsilon_{Lamb} = \exp\left(-\frac{r^2}{\sigma^2}\right) = f(r, \sigma) \quad (2.54)$$

and

$$\epsilon_{blob} = 1 - \left(\frac{r^2}{r^2 + \delta^2}\right) = 1 - f(r, \delta) . \quad (2.55)$$

Self-similar velocity profiles of a potential, blob and Lamb-Oseen vortex models are given in Fig. 2.4, where one can see how the velocity profiles compare to the potential solution, which is representative of an inviscid flow. The other two models represent physical aspects present in a realistic vortex core where diffusion is naturally present. The profiles shown in the figure are obtained for  $\sigma = \delta$  and, for these particular parameters, one can observe that the Lamb vortex model is more singular than the vortex blob.

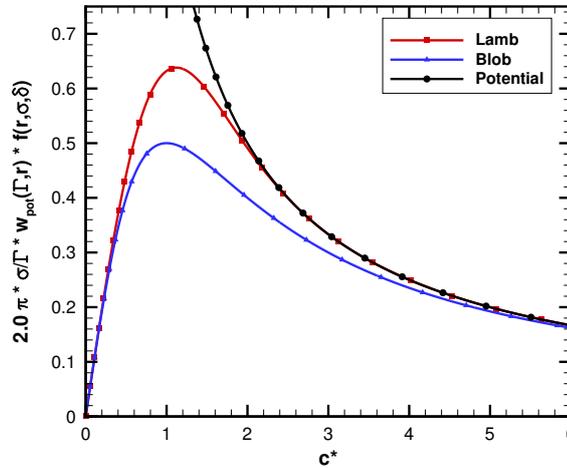
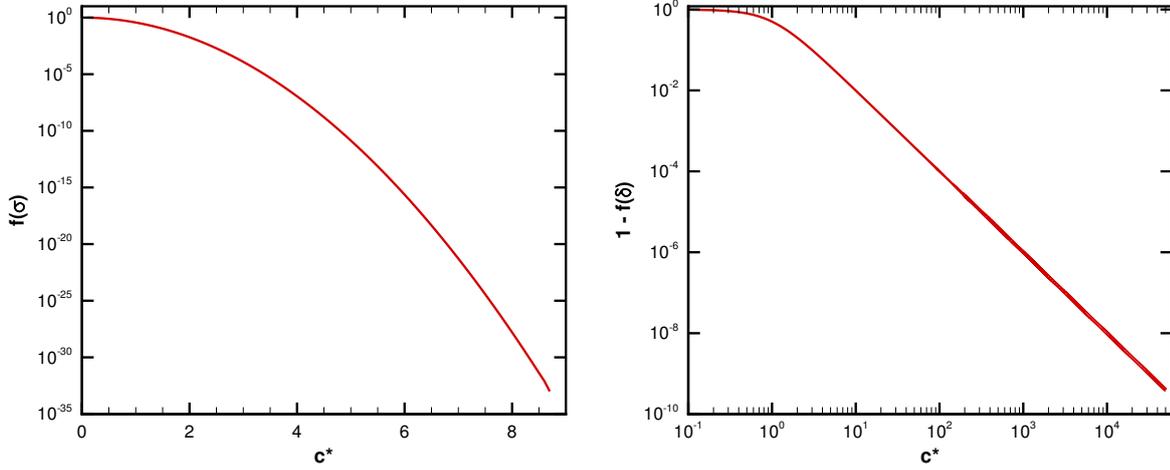


Figure 2.4: Self-similar velocity profiles for different kinds of vortex models.

From Fig. 2.5, one can see that the desingularization parameter, Eqs 2.54 – 2.55, decays exponentially for the Lamb vortex and algebraically with  $O(1/c^{*2})$  for the blob. The latter shows a slow decay which indicates a non-compact vorticity distribution, being the opposite of the Lamb model.



(a) Lamb vortex.

(b) Vortex blob.

Figure 2.5: Decay of the smoothing parameter.

### 2.2.5 Time marching method

In the Lagrangian inviscid problem, after computing the local velocity flow field at each particle, it is only necessary to move them. In order to compute the position of the individual particles, it is necessary to solve the following differential equation:

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t) = F(\omega, \phi, \mathbf{x}, t) . \quad (2.56)$$

Any explicit ordinary differential equation solver, such as Adams-Bashforth or Runge-Kutta schemes, can be used to solve Eq. 2.56. In the present work, only the first, the second and the fourth order Runge-Kutta schemes are analyzed and employed.

The first order Euler's method, RK1, computes the position  $\mathbf{x}$  at step  $n + 1$  based on the present position at  $n$ , interval  $\Delta t$ , and also on the instantaneous velocity field  $\mathbf{u}_n$ :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t F(\omega, \phi, \mathbf{x}, t) , \quad (2.57)$$

where the velocity field is evaluated once since no sub-steps are employed:

$$F(\omega, \phi, \mathbf{x}, t) = \mathbf{u}_n . \quad (2.58)$$

The second order Heun's method, RK2, requires two evaluations of the velocity field, since one sub-step  $\tilde{x}_{n+1}$  is also calculated:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{\Delta t}{2} F(\omega, \phi, \mathbf{x}, t) , \quad (2.59)$$

with

$$F(\omega, \phi, \mathbf{x}, t) = \mathbf{u}_n + \tilde{\mathbf{u}}_{n+1} , \quad (2.60)$$

and

$$\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{u}_n . \quad (2.61)$$

The classical fourth order Runge-Kutta method, RK4, requires four evaluations of the velocity field since three sub-steps are present:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{\Delta t}{6} F(\omega, \phi, \mathbf{x}, t) , \quad (2.62)$$

where

$$F(\omega, \phi, \mathbf{x}, t) = \mathbf{u}_n + 2\hat{\mathbf{u}}_{n+\frac{1}{2}} + 2\tilde{\mathbf{u}}_{n+\frac{1}{2}} + \bar{\mathbf{u}}_{n+1} . \quad (2.63)$$

In the equations above, the terms  $\hat{\mathbf{u}}$ ,  $\tilde{\mathbf{u}}$  and  $\bar{\mathbf{u}}$  represent intermediate velocity fields which are computed at sub-iterations of the Runge-Kutta schemes, based on intermediate positions  $\hat{\mathbf{x}}$ ,  $\tilde{\mathbf{x}}$  and  $\bar{\mathbf{x}}$ . This sub-steps are calculated according to

$$\hat{\mathbf{x}}_{n+\frac{1}{2}} = \mathbf{x}_n + \frac{\Delta t}{2} \mathbf{u}_n , \quad (2.64)$$

$$\tilde{\mathbf{x}}_{n+\frac{1}{2}} = \mathbf{x}_n + \frac{\Delta t}{2} \hat{\mathbf{u}}_{n+\frac{1}{2}} , \quad (2.65)$$

and

$$\bar{\mathbf{x}}_{n+\frac{1}{2}} = \mathbf{x}_n + \Delta t \tilde{\mathbf{u}}_{n+\frac{1}{2}} . \quad (2.66)$$

Single-step methods like the  $k$ th-order Adams-Bashforth are also useful since they evaluate once the velocity field per time step; also, they still provide an error proportional to  $O(\Delta t^k)$ . However, they require the storage of previous velocity fields which can introduce further problems with respect to the creation of vortices. Moreover, the DVM has no intrinsic numerical dissipation and, then, the presence of spurious roots which appear in this class of time-marching schemes can be also a problem since they are not dissipated. These methods can be applied in classical CFD algorithms since their numerical dissipation can handle the spurious roots. It is important to mention here that Runge-Kutta schemes do not introduce spurious roots.

## 2.2.6 Stability in time marching method

In grid-based computational fluid dynamics methods, the major parameter driving numerical stability and time accuracy based on grid spacing is the Courant-Friedrichs-Lewy number, CFL (Lomax *et. al*, 1999). However, this parameter is not definable in a meshless method and, therefore, the time step is not easily determined. Hence, tests are performed in order to understand the influence of time step in the results of a simulation using the DVM.

Previous work involving problems with absence of viscosity or boundaries are based in the computation of the Hamiltonian, an operator that indicates the energy conservation in the system, as a metric to be used for accuracy. The Hamiltonian, when  $i \neq j$ , is measured as

$$H = \frac{1}{2\pi} \sum_{i=1}^N \sum_{j=1}^N \Gamma_i \Gamma_j \log(r_{ij}) . \quad (2.67)$$

For more details, one should see Milinazzo and Saffman (1977), Kantelis and Widnall (1986), Krasny (1986b, 1987) and Sparlat (1988).

Other parameters can be measured to assess the time step influence. For viscous flows, the angular impulse,  $\bar{A}$ , decays while the linear impulses,  $I_x$  and  $I_y$  along  $x$  and  $y$  directions, should be conserved. On the other hand, linear and angular impulses are invariants for inviscid flows in an unbounded domain. These are given by:

$$I_x = + \sum_{i=1}^N \Gamma_i y_i , \quad (2.68)$$

$$I_y = - \sum_{i=1}^N \Gamma_i x_i , \quad (2.69)$$

$$\bar{A} = -\frac{1}{2} \sum_{i=1}^N \Gamma_i (x_i^2 + y_i^2) . \quad (2.70)$$

An investigation of this issue is performed with a simple but illustrative way to understand the effect of low accuracy in the time marching scheme. A large vortical structure discretized by  $N$  vortex particles is marched forward in time, where the interaction among vortex particles creates a rotational motion of the structure. The algorithm proposed by Box and Muller (1958), Eqs. 2.71 – 2.74, generates the initial position. The particles are randomly distributed at  $x_i$  and  $y_i$ , following a Gaussian shape with standard deviation  $\tilde{\sigma} = 0.1$ . The random numbers  $P$  and  $Q$  are generated in the range between 0 and 1.

$$R = \tilde{\sigma} \sqrt{-2 \log(P)} , \quad (2.71)$$

$$\theta = 2\pi Q , \quad (2.72)$$

$$x_i = R \cos(\theta) , \quad (2.73)$$

$$y_i = R \sin(\theta) . \quad (2.74)$$

Tests are performed using  $n = 30k$  vortices with viscous cores  $\sigma = 0.001$ . The total circulation is set as  $\Gamma = 1.0$  and  $\Gamma = 10.0$ , which is equally divided among all  $N$  particles.

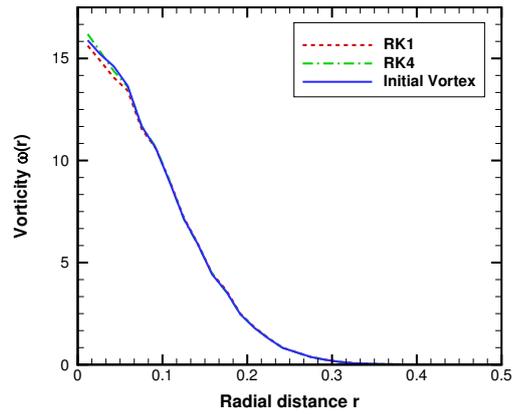
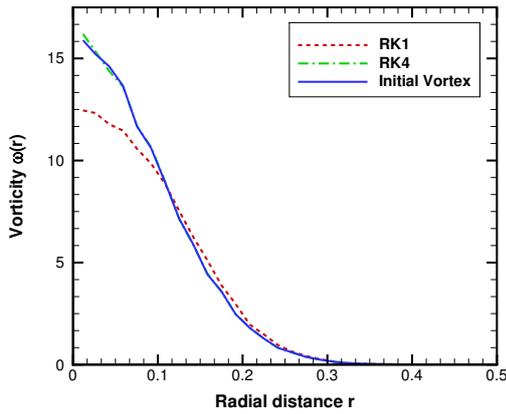
Following the example from Lewis (1991), the radial distribution  $n_j$  vortex particles, inside an annulus  $j$  defined by  $r_j$  to  $r_{j+1}$ , from  $j = 1$  to  $j_{max}$ . These annulus are around the center of the vortex, placed from  $r_1 = 0.0$  up to most outside contour at  $r_{j_{max}} = 0.5$ .

$$\omega(r) = \frac{n_j}{N\pi(r_{j+1}^2 - r_j^2)} , \quad (2.75)$$

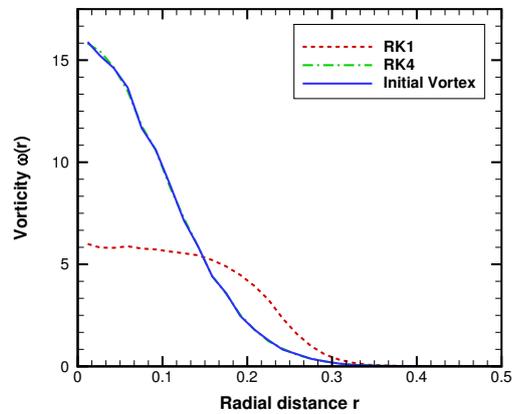
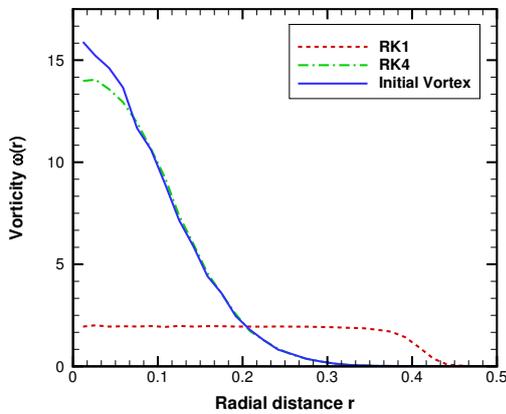
Results using 30 annulus are plotted in Fig. 2.6 based on the root-mean-square radius (RMS), given by:

$$r_{rms} = \sqrt{\frac{1}{2}(r_j^2 + r_{j+1}^2)} . \quad (2.76)$$

One should expect a constant distribution of elements for all time steps since there is no radial velocity. For a dimensionless time  $t^* = 0.5$ , Fig. 2.6 shows the vorticity for the first and fourth order Runge-Kutta schemes, RK1 and RK4. The solutions are compared to the initial solution of the large vortex, indicating that a low accuracy in the time marching method tends to spread the distribution, reducing the kinetic energy of the particles and adding numerical diffusion to the solution. More pronounced differences among the solutions are observed for the  $\Gamma = 10.0$  problem. Results of Eqs. 2.67 and 2.70 are shown in Fig. 2.7, where one can see that the RK1 is non-conservative and the RK4 obtains accurate solutions.

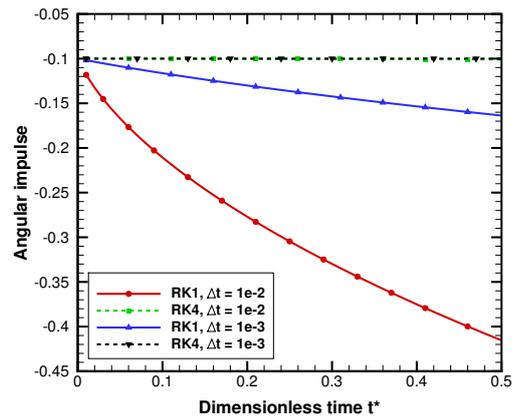
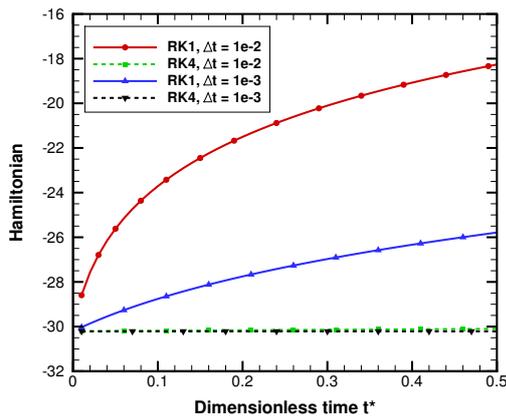


(a) 50 steps,  $\Delta t = 0.01$ ,  $\sigma = 0.001$ ,  $\Gamma = 1.0$ . (b) 500 steps,  $\Delta t = 0.001$ ,  $\sigma = 0.001$ ,  $\Gamma = 1.0$ .



(c) 50 steps,  $\Delta t = 0.01$ ,  $\sigma = 0.001$ ,  $\Gamma = 10.0$ . (d) 500 steps,  $\Delta t = 0.001$ ,  $\sigma = 0.001$ ,  $\Gamma = 10.0$ .

Figure 2.6: Analysis of the time marching schemes.



(a) Hamiltonian,  $\Gamma = 10$ .

(b) Angular impulse,  $\Gamma = 10$ .

Figure 2.7: Analysis of the invariants.

## 2.3 Viscous diffusion step

Viscosity is a natural mechanism which reduces gradients. When a fluid property has a large concentration in a flow region, it tends to be spread out by viscous diffusion according to the Reynolds number. It acts on solid walls through the no-slip boundary condition, where the fluid attaches to the surface and moves with its velocity. This way, there is production of vorticity along the walls. Furthermore, the rate of creation depends on the pressure gradient along the walls and, in some cases, vorticity may even be destroyed. Combining these phenomena, Eq. 2.25 can be interpreted as:

$$\frac{\partial \omega}{\partial t} = \text{Diffusion} + \text{Production} - \text{Destruction} \quad (2.77)$$

In the DVM, the property *vorticity* can be interpreted in two ways: a local property inside each particle, which avoids the singular velocity field of a potential vortex (see section 2.2.4), or the whole field represented by all vortices elements.

### 2.3.1 Diffusion models

Chorin (1973) was the first to propose a method to solve the diffusion process, the Random Walk Method (RWM). The vortex particles have a random displacement in order to simulate the macroscopic diffusion of the larger vortical structures due to Brownian motion of molecules. The random displacement of each vortex in  $x$  and  $y$  axis depends on both  $P$  and  $Q$ , which are random numbers between 0 and 1, is given by:

$$\Delta x_i = \sqrt{\frac{4\Delta t}{Re} \log\left(\frac{1}{P}\right)} \cos(2\pi Q) , \quad (2.78)$$

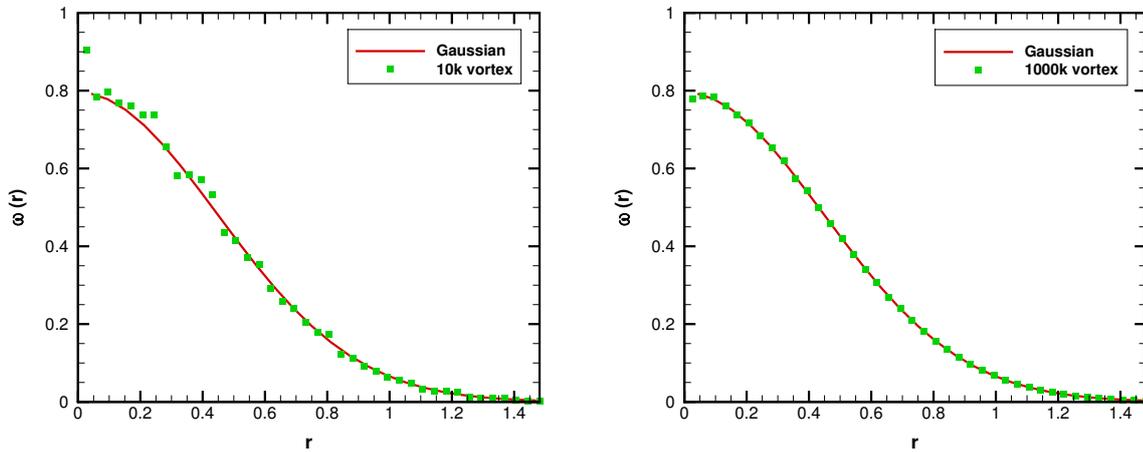
$$\Delta y_i = \sqrt{\frac{4\Delta t}{Re} \log\left(\frac{1}{P}\right)} \sin(2\pi Q) . \quad (2.79)$$

The vorticity diffusion of the larger vortical structures, which are composed by several vortex particles, follows an exponential decay by a Gaussian distribution whose shape is given by:

$$\omega(r, t) = \frac{Re\Gamma}{4\pi t} \exp\left(-Re\frac{r^2}{4t}\right) . \quad (2.80)$$

Following the analysis performed by Lewis (1991) to illustrate the method, 10k vortex particles are placed at the origin and diffused with  $Re = 1$  and  $\Gamma = 1.0$  during a non-dimensional time  $t = 0.1$  and the results are shown in Fig. 2.8(a). Increasing the number of elements to 1000k, Fig. 2.8(b), shows a convergence to the Gaussian solution given by Eq. 2.80. As both convection and diffusion errors are reduced for  $N \rightarrow \infty$ , the use of fast summation algorithms is indispensable.

Figure 2.9 shows results of the diffusion model applied to different Reynolds numbers. Numerically, it shows a good representation for all Reynolds numbers, however, it is conceptually wrong to use a stochastic method for laminar flows at small Reynolds numbers, or to use a two-dimensional algorithm for a turbulent flow in three-dimensions at high Reynolds numbers. For further review on the model, one should see Milinazzo and Saffman (1977); Ghoniem (1985).



(a) 10k particle vortex,  $Re = 1$ .

(b) 1000k particle vortex,  $Re = 1$ .

Figure 2.8: Convergence of diffusion process using the random walk method.

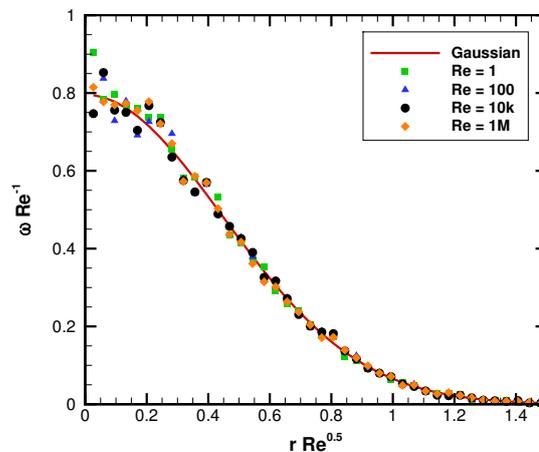


Figure 2.9: Diffusion for different  $Re$  using 10k vortex particles.

Another model for the diffusion process is the Core Spreading Method, Eq. 2.81, in which the viscous core of the Lamb-Oseen vortex grows indefinitely in time, spreading the vorticity. This way, the induced velocity becomes smaller as the vortex diffuses reducing its kinetic energy.

$$\sigma = \sqrt{\frac{4t}{Re}}. \quad (2.81)$$

Leonard (1980) shows that the error of DVM is proportional to  $O(\sigma^2)$ . Hence, if the core grows indefinitely, one loses control of the error in the method. Also, Greengard (1985) proved that the convergence with this diffusion scheme leads to wrong results. Later, Rossi (1996) proposed corrections to the method in order to improve it. In these corrections, the vortex grows in time until it becomes too large and splits in smaller vortices. Although conceptually good and also related to the energy cascade of scales in turbulence, this new method involves arbitrary numerical parameters which penalizes its application. Also, vortex splitting considerably increases the computational cost of an  $O(N^2)$  method.

These are the simplest ways to solve the diffusion mechanism without depending on a computational mesh. Other schemes that use computational grids are available in the literature. However, this mesh usage is against the logic of a Lagrangian method. For more details on other methods, one should see the work of Shiels (1998); Barba (2004); Koumoutsakos (1993); Shankar and Dommelen (1996); Ogami and Akamatsu (1991).

### 2.3.2 Vorticity generation

Vorticity generation occurs along the solid walls due to the no-slip boundary condition in a viscous flow. To mimic these effects, new vortex particles are created and convected every time step, with the possibility to quickly reach hundreds of thousands of particles in a simulation, which is very unfavorable for an  $O(N^2)$  method. The generation of vorticity is not a consensus among authors in literature. Some models are adjusted by experiments or boundary layer parameters, and, although physically well-based, they lose generality (Tsutsui *et. al*, 1997). Other methods depend partially on computational grids or redistribution of the vortex elements and, therefore, they go against the Lagrangian nature of the problem. Using BEM, the panels can have distributions of sources or vorticity layers and the creation process is different for each distribution of singularities. In this work, only the source distributions are implemented.

For a source distribution  $\sigma(\mathbf{x})$  in the BEM, the discrete vortex  $\Gamma$  is placed at a fixed position above the panel, leading to a matrix  $M(2n,2n)$ , for  $n$  being the number of panels. The vector at the right side of the equation has to be evaluated every time step, based

on normal and tangential velocities induced by the wake and the incident flow on each panel centroid. For a rigid body, the inverse matrix can be calculated only once and stored, such that the solution is simply a matrix-vector product, with computational cost  $O(n^2)$ . Furthermore, the solution is exact and not with tolerance as obtained by iterative methods. The result of this equation is the discrete vortex circulation and source intensity per length, given by:

$$\begin{Bmatrix} \Gamma \\ \sigma \end{Bmatrix} = \begin{bmatrix} \mathbf{M} \end{bmatrix}^{-1} \begin{Bmatrix} -\mathbf{u} \cdot \mathbf{n} \\ -\mathbf{u} \cdot \mathbf{t} \end{Bmatrix} \quad (2.82)$$

The circulation of a body is always null for source distribution. Hence, it is only necessary to enforce mass conservation to satisfy continuity equation, Eq. 2.1. Given a length  $S$  for the panel  $i$ , the mass conservation is written as

$$\sum_{i=1}^n \sigma_i S_i = 0 \quad (2.83)$$

When vorticity generation is present at boundaries, the time step may be based on the panel length and the characteristic velocity of the flow to avoid a vortex to cross several panels (for large  $\Delta t$ ) or to cluster on the vicinity of its "birth-place" (for small  $\Delta t$ ). For more details, one should see the work of Koumoutsakos (1993), Shiels (1998), Kamemoto (2004), Lewis (1991) and Barba and Cooper (2009).

### 2.3.3 Vorticity destruction

Another detail when simulating solid surfaces is that, as the viscous core is finite, the vorticity can be partially inside the body or, in the worst case, it can be totally moved to the interior of the body. This happens due to finite-time steps or finite velocities on the panels edge, due to the limitations on BEM collocation points. In these cases, it is possible to eliminate those vortices inside the body and to redistribute their vorticity in the newborn vortex particles in order to satisfy the conservation of circulation. Also, some authors treats the vortex particles as perfectly elastic circles and bounce them back into the fluid in case of collision, based on their incidence angle relative to the body surface.

Other consideration is that when two vortex-particles come close to each other, it is possible to merge them in a single particle, and both first and second moments are conserved. The main benefit is to reduce the total number of elements and speed-up computations. Some considerations have to be done, which includes the point where vortex can start to merge, since one desire good resolution close to the body. Also, vortex far away have little influence in the body, so their merging does not compromise the solution or calculation of pressure.

For details on vorticity generation and destruction, one should see the work of Chorin (1978); Porthouse (1983); Koumoutsakos *et. al* (1994); Shiels (1998); Kamemoto (2004); Ramachandran *et. al* (2007); Barba and Cooper (2009).

### 3 FAST MULTIPOLE METHOD

The fast multipole method, FMM, was developed by Greengard and Rokhlin (1987) for the solution of  $N$ -body problems and it is listed as one of the top 10 algorithms of the twentieth century (Cipra, 2000). The method employs a divide-and-conquer strategy which reduces the computational cost of the  $N$ -body problem from  $O(N^2)$  operations to  $O(N)$  by the interaction among clusters of sources and observers.

The method has extensive applications in boundary element methods, BEM, and, in this case, it can be applied for the solution of potential flows, heat transfer and acoustic scattering around arbitrary geometries. The FMM can also be employed together with the method of moments (MoM) to accelerate the solution of problems of computational electromagnetism involving potential fields from electric charges. Other applications of the FMM include problems of classical mechanics involving gravitational potential from celestial bodies and Lagrangian methods in fluid flow simulations. In the latter case, one can cite the acceleration of the discrete vortex method, DVM, as a possible application.

The main idea of the FMM is to use a series expansion for the Green's function that governs the interaction between an observer and a source term, respectively  $\mathbf{x}$  and  $\mathbf{y}$  in Eq. 3.1. The error of the method can be controlled according to the truncation of the series expansion which has the form:

$$G(\mathbf{x}, \mathbf{y}) = \sum_0^{\infty} O(\mathbf{x})S(\mathbf{y}) . \quad (3.1)$$

In the equation above, the term  $O(\mathbf{x})$  represents a function which depends solely on the observer position and  $S(\mathbf{y})$  represents a function dependent on the source position and its intensity.

Usually, using Eq. 3.1, one must use a power series where the number of terms in the series gives the accuracy in the approximation as shown in Fig. 3.1, where each curve is given by a different truncation point. The power series converges for  $|\mathbf{x} - \mathbf{y}| \rightarrow \infty$  but it does not converge when the source is close to the observer,  $|\mathbf{x} - \mathbf{y}| \rightarrow 0$ . Thus, the series expansion fails requiring a direct evaluation of the Green's function for adjacent sources and observers. For instance, the function  $f(x) = e^{-1/x^2}$  can be approximated as

$$f(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{-2n}}{n!} . \quad (3.2)$$

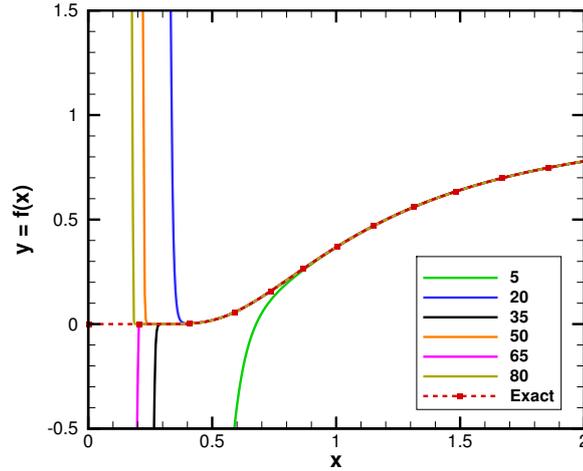


Figure 3.1: Power series expansion of example function  $f(x) = e^{-1/x^2}$ .

A simplified scheme of the FMM algorithm is shown in Fig. 3.2 where the computational domain involves all the elements (sources and observers). As can be observed, this computational domain is further divided in order to create smaller boxes that separate the elements, defining clusters of sources or observers. The center of these clusters are indicated by the orange dots, where it is placed an equivalent charge (multipole expansion) that considers the influence of all elements inside its box. Then, the multipole expansions are used to evaluate interaction of sources at distant locations (centroids of far away boxes). Hence, by the translation to the centroids of observer boxes, one can create local representation of the sources at the observer position. The first operation, (A), is shown in orange lines in Fig. 3.2(a) while the second operation, (B), is represented by black lines in the same figure.

After that, a reverse operation translates the influences from the sources, computed by operations (A) and (B), to all the individual observers surrounded by a box, operation (C) shown by blue lines in the figure. As the method is not applicable among near-field clusters of particles, these calculations are still performed directly solving the Green's function, operation (D) represented by red lines in Fig. 3.2(b). In order to obtain convergence of the power series, the box separation criterion is, at least, the length of a source box (see Greengard and Rokhlin (1987) for more details on this criterion).

The far-field operations (A–C) of the FMM are responsible for reducing the computational cost of the method to  $O(N)$  applying the divide-and-conquer strategy. However, since close range influences are still performed directly by operation (D), a fraction of the total computational time depends on the square of the number of particles in each cluster,  $O(N^2)$ .

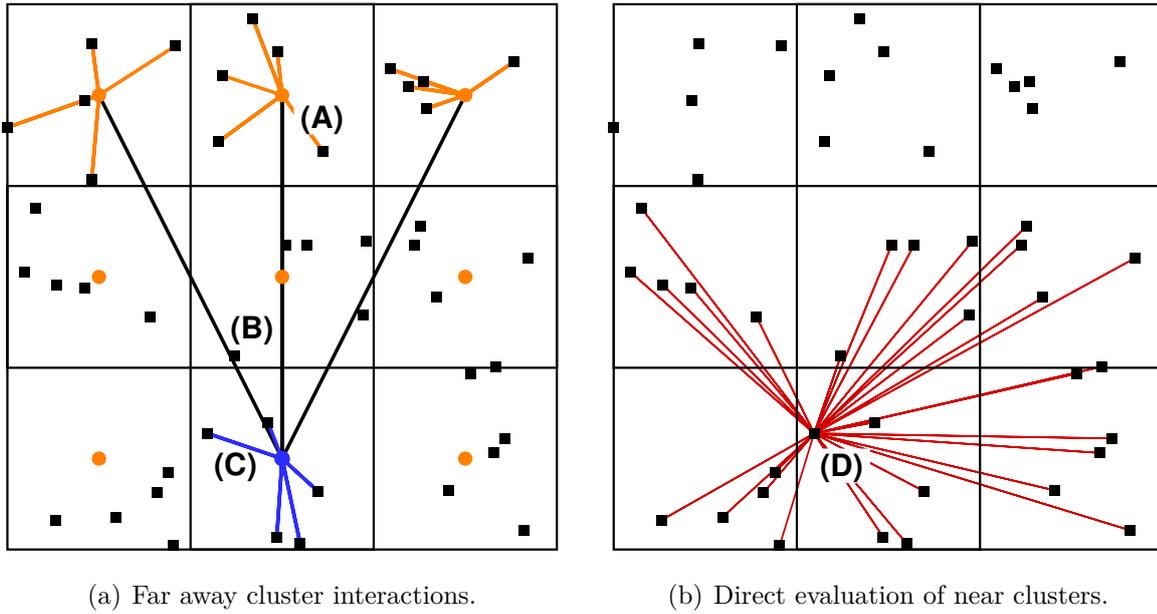


Figure 3.2: Interactions in the near-field and far-field.

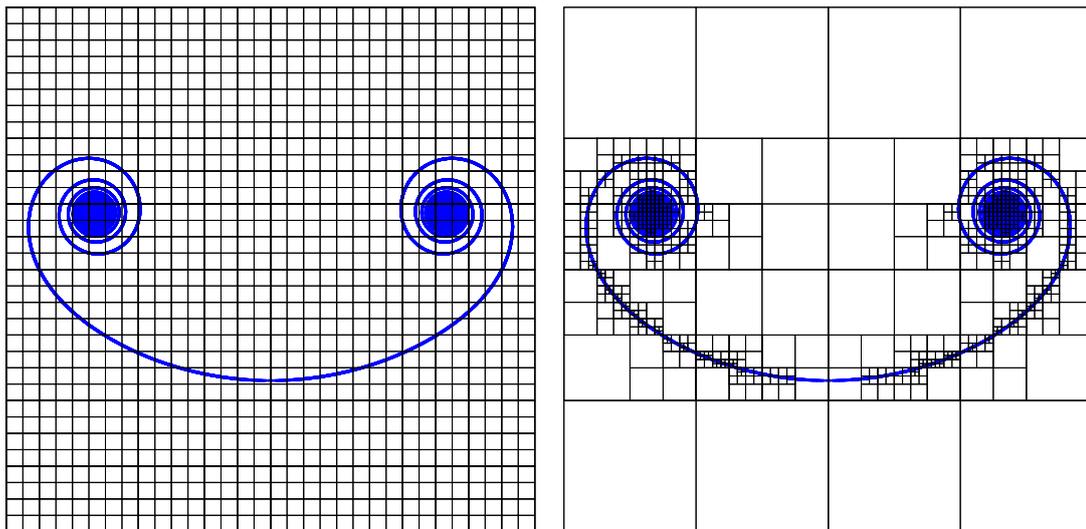
The spatial arrangement of the sources and observers plays an important role in the FMM algorithm and two different methodologies are proposed to handle the problem of box generation. The first one is called the global refinement algorithm, and its maximum efficiency is obtained when the distribution of sources and observers is nearly uniform in the domain (Greengard and Rokhlin, 1987). In this case, the entire FMM computational domain is uniformly refined. In this refinement, parent boxes at level  $\ell - 1$  are always divided into 4 identical children boxes at level  $\ell$ , despite the fact that some boxes may be empty or that the parent box at level  $\ell - 1$  may have few particles. This process is performed to a prescribed maximum level  $L$  in which the near-field and far-field computations have nearly the same computational cost. If the boxes are overcrowded, the near-field calculations become expensive and the balance between far and near-field computations is lost, increasing computational time. On the other hand, if too many boxes are used in a specific refinement level, the far-field cost becomes too high.

The second methodology, which is called the adaptive algorithm, is focused in reducing computational time by employing an efficient division of the domain (Carrier *et. al*, 1988). Regions with high-density of elements are more refined in order to reduce direct summation, while sparse distributions have coarser refinements to reduce far-field operations cost. Only boxes containing several elements are further refined, which means that the empty ones are neglected. This process is performed until all boxes reach the maximum prescribed number of particles. This number is specific of each problem and it is determined by numerical inspection. Therefore, the spatial refinement of the computational domain depends on the cloud arrangement.

Although the computational cost of both methodologies depends on the number of elements per box, the refinement in the first method is based on the average number of elements per box. The second method treats each box individually in order to reduce the number of elements in overcrowded boxes and, thus, reducing the near-field computational cost. Moreover, it also avoids the presence of boxes at high refinement levels with few elements, reducing the far-field computational cost.

Both methods rely on the construction of lists of boxes that control the interactions among clusters of elements, and these lists are related only to the boxes, not to the particles. The global method requires a list of neighbor boxes and a list of interaction boxes in the far-field. Both lists are built at each specific level. The adaptive method uses four lists with neighbor and far-field boxes which are not necessarily in the same level, so this increases the complexity of the method. However, the motion of the particles in the adaptive case changes the FMM refinement throughout the FMM computational box. So, every time step the adaptive refinement must be performed again to create the boxes, while the global refinement avoids this constant preprocessing.

In Fig. 3.3 there is an example of the application of both methods showing the refinement of the computational box in a problem solved using the discrete vortex method. In this figure, an airplane wake is modeled by discrete vortices given by the Trefftz plane problem. More details about this problem are provided in Chapter 4. As one can see, some boxes contain several particles, while others are empty. This way, the global method, shown in 3.3(a) is not the best option, while the adaptive method, 3.3(b), is more suitable since it perform the cluster-to-cluster operations more efficiently.



(a) Global method.

(b) Adaptive method.

Figure 3.3: Different types of refinement in the FMM.

## 3.1 Multi-level FMM

In this section, the multi-level implementation of the FMM algorithm is explained for the global refinement approach.

### 3.1.1 Preprocessing

The hierarchical multi-level model proposed by Greengard and Rokhlin (1987) uses the global refinement approach and is suited for cases with uniform distributions of vortex elements. The computational domain is a square box with side  $S_0$ , at level  $\ell = 0$ , containing all the  $N$  vortex elements, shown in Fig. 3.4(a). The refinement of the initial domain is performed dividing this box into four identical boxes at level  $\ell = 1$ , Fig. 3.4(b). This division is recursively repeated throughout the next levels, as shown in Fig. 3.4(c), until a prescribed maximum level  $L$ , independently of the number and distribution of elements. Generalizing this recursive step, a box at generic level  $\ell$ , called *parent*, is further divided in other four boxes at level  $\ell + 1$ , called *children*. Since all boxes are divided, the genealogical tree that represents the hierarchical relation of all boxes is complete. This leads to a number of boxes at level  $\ell$  equal to  $4^\ell$ .

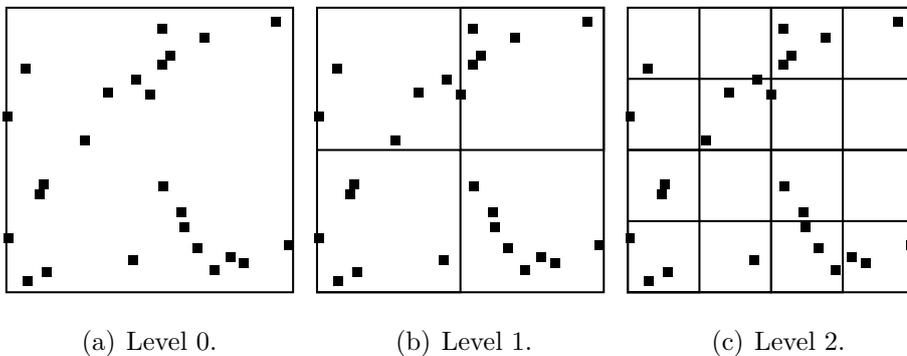


Figure 3.4: Refinement of initial domain from level  $\ell = 0$  to  $\ell = 2$ .

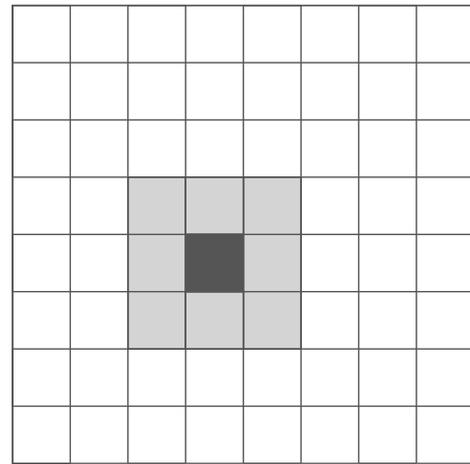
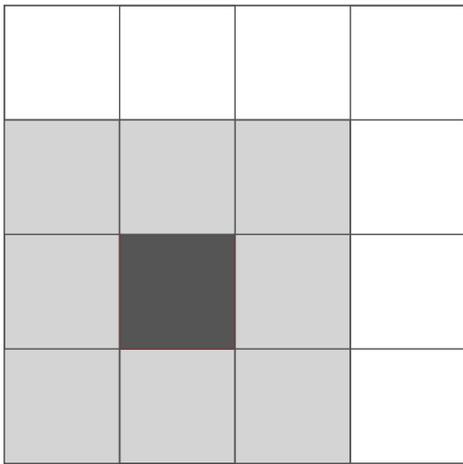
The relations among boxes at the same level are given by two lists containing neighbors and far-field interactions. The former contains those boxes with at least a common node with box  $b$ , and  $b$  itself, since they do not satisfy the separation criterion of the FMM. The latter contains those boxes which are *well-separated* from  $b$  and that are allowed to interact with  $b$  via the FMM. The term *well-separated* is used for boxes which are at least one box length distant from  $b$ .

A recursive scheme to create both lists quickly is employed beginning in level 0. The box  $b$  at level  $\ell$  has a parent box at level  $\ell - 1$  called  $B$  which has up to 8 adjacent boxes

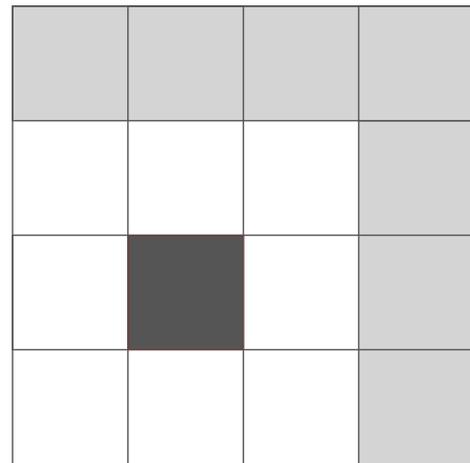
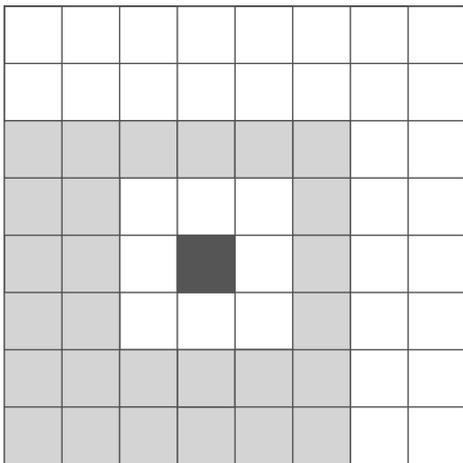
with at least one node in common, as shown in Fig. 3.5(a). These 8 boxes plus  $B$  itself have 4 children each with a total of 36 children boxes given by:

- $b$  itself (dark gray box in Fig. 3.5(b))
- 8 neighbors of  $b$  with at least one common node with  $b$ , being 5 children of  $B$ 's neighbors and 3 children of  $B$  (light gray boxes in Fig. 3.5(b)),
- 27 boxes in interaction list of  $b$ , which are children of  $B$ 's neighbors and are not neighbors of box  $b$  (light gray boxes in Fig. 3.5(c)).

The influence at  $b$  from further boxes, *i.e.*, those that are children of the boxes in  $B$ 's interaction list, Fig. 3.5(d), is computed hierarchically through  $B$ , characterizing a multi-level scheme which speeds up computations.



(a) Level 2 neighbors (light gray) of  $B$  (dark gray). (b) Level 3 neighbors (light gray) of  $b$  (dark gray).



(c) Interaction list (light gray) of  $b$  (dark gray) at level 3. (d) Interaction list (light gray) of  $B$  (dark gray) at level 2.

Figure 3.5: Representation of interaction and neighbor lists for different levels in FMM.

The preprocessing step also allows the computation of mathematical operations and storage of data to further speed-up calculations during the FMM execution. These operations includes the calculation of Newton's binomial coefficients, centers of the boxes where clusters of particles are placed, and also a third list of boxes to indicate the four children of a parent box. The preprocessing step can be resumed in five main operations:

1. calculation of Newton's binomial coefficients depends only on the series truncation,
2. calculation of boxes' centers depends both on refinement level and FMM domain size,
3. creation of parent-child lists depends only on maximum refinement level,
4. creation of neighbor's lists depends only on maximum refinement level,
5. creation of far-field interaction lists depends only on maximum refinement level.

Furthermore, for the global method, these operations are not strongly dependent on the particles position. The unique dependency is that the FMM domain has a coverage of all elements. Hence, in the dynamic problem of the discrete vortex method, the global refinement scheme is less sensitive compared to the adaptive one.

### 3.1.2 Global refinement

After the preprocessing, the computation of the interactions among sources and observers is performed by the fast multipole method following the steps below:

- Step 1, after the refinement and creation of all boxes up to the maximum level, one must map the  $N$  source and observer elements inside all boxes in the finest level. If the observers do not coincide with the sources, two mapping steps are necessary.
- Step 2, one must create the multipoles at the finest level in a process called particle-to-multipole (P2M). It consists in grouping the influences from vortex elements inside a box to its center using the FMM series expansion.
- Step 3, the clustering of existing multipoles is performed from children boxes to the parent's box, creating multipole expansions of larger clusters of particles. This operation is called multipole-to-multipole (M2M) and it is performed recursively from the finest level of refinement up to level 2 for free-space domains (since there are no operations among level 0 or 1 boxes) or up to level 0 for periodic domains (in this case, there are operations among boxes in both level 0 and 1). Empty children boxes can be neglected in this step, reducing computational cost.

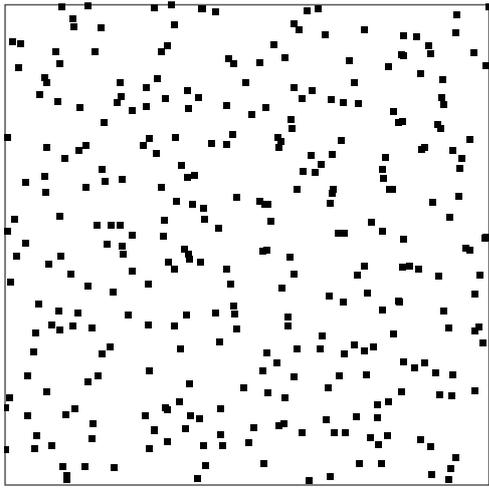
- Step 4, at the tree top, interactions of the largest clusters at level 0 are computed for periodic domain problems. For a finite domain, this step can be skipped. Special attention is required by both adjacent boxes of the central domain in periodic problems since their interactions are performed using the same ideas of step 4 below since they are not *well-separated*.
- Step 5, a combination of two operations occur: interactions from well-separated source clusters to an observer are computed at the same level,  $\ell$ . Also, translations of local representations from the observer cluster to its children at level  $\ell + 1$  for creation of local expansions in the next level, accounting its parent's far-field interactions. The former operation is called multipole-to-local (M2L) while the latter is local-to-local (L2L). These operations repeat recursively up to the finest level.
- Step 6, with all local representations evaluated, one translates the influences of the far-field sources at the centers of finest boxes to all local observers. This operation is called local-to-particle (L2P).
- Step 7, near-field interactions among particles are evaluated directly at the finest level of refinement for the boxes contained on the neighbors list and summed to the far-field interactions. This final operation is particle-to-particle (P2P).

To illustrate the steps above, Fig. 3.6(a) shows the distribution of vortex particles along the initial domain, which is refined up to the prescribed maximum level  $L$ . In the present example,  $L = 3$  as shown in Fig. 3.6(b).

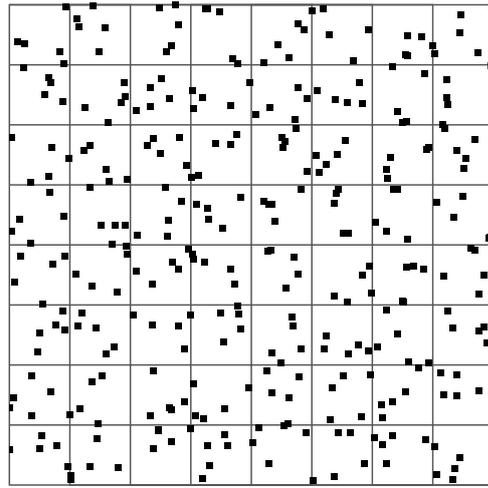
The creation of multipoles at the finest level in the P2M step is shown in Fig. 3.7(a), resulting in equivalent charges placed at the box centers in Fig. 3.7(b). Larger clusters at level  $\ell - 1$  are created from smaller ones at level  $\ell$  using M2M operations as shown in Fig. 3.7(c) and (d).

Figure 3.8 shows the downward pass and near-field interactions. In Fig. 3.8(a), one can see the M2L interactions at level  $\ell - 1$ . This operation is followed by the L2L steps which translate the influences from far-away clusters to children at level  $\ell$  in Fig. 3.8(b), and the M2L operations at level  $\ell$  in Fig. 3.8(c). In Fig. 3.8(d), one can see the calculation of the far-field induced velocity effects from a translation of the local influence from the center of the box to the vortex elements inside the box, L2P.

Finally, Fig. 3.9 shows the final particle-to-particle calculations performed directly through the Biot-Savart law to evaluate the near-field induced velocity.

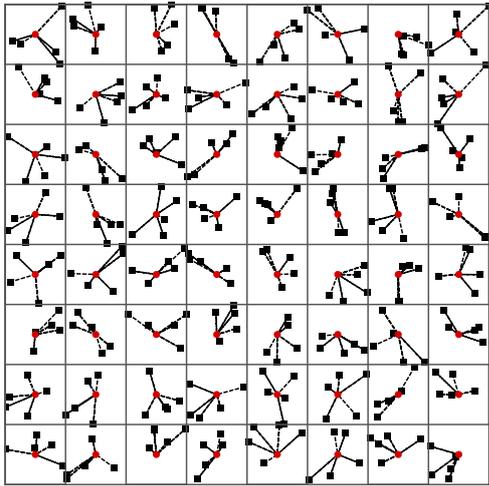


(a) Particle distribution and initial domain.

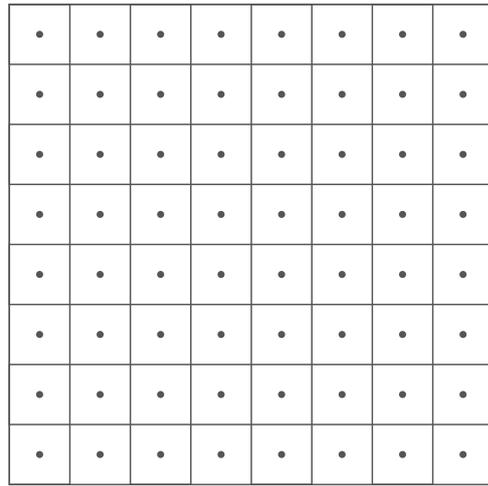


(b) Refinement of the initial domain.

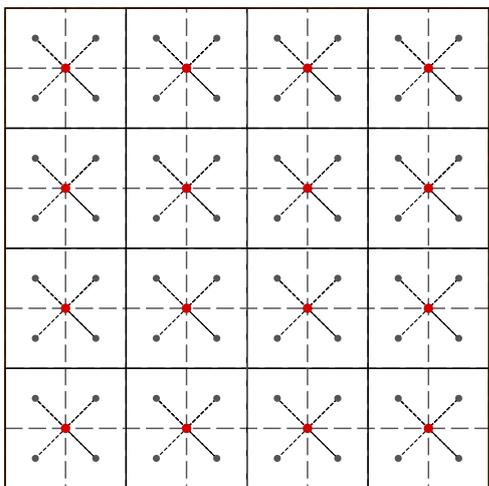
Figure 3.6: Box creation in preprocessing.



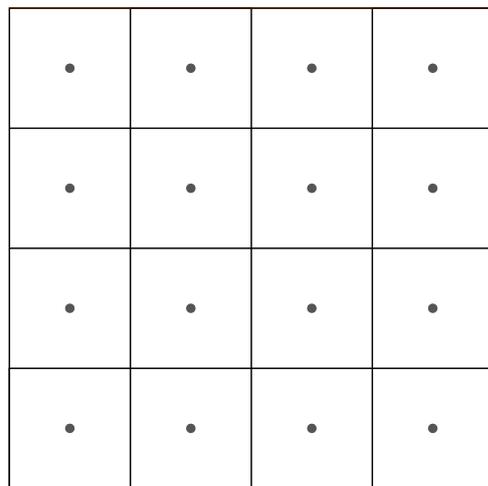
(a) Particle-to-multipole at the finest level.



(b) Equivalent charges in boxes at level 3.

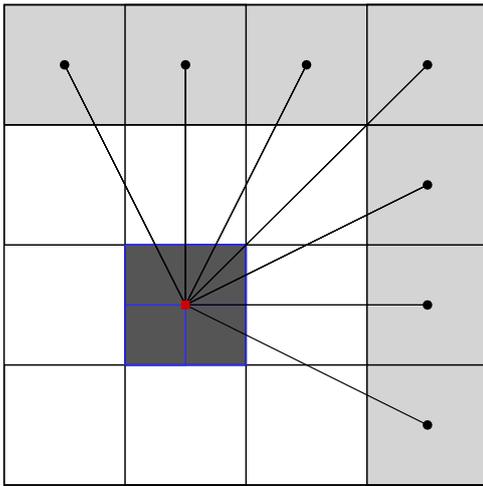


(c) Multipole-to-multipole from level 3 to 2.

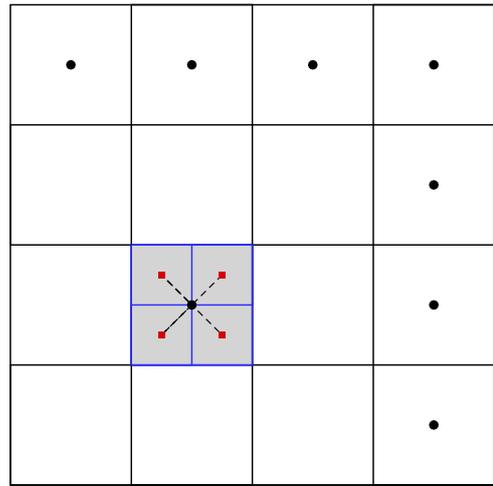


(d) Equivalent charges in boxes at level 2.

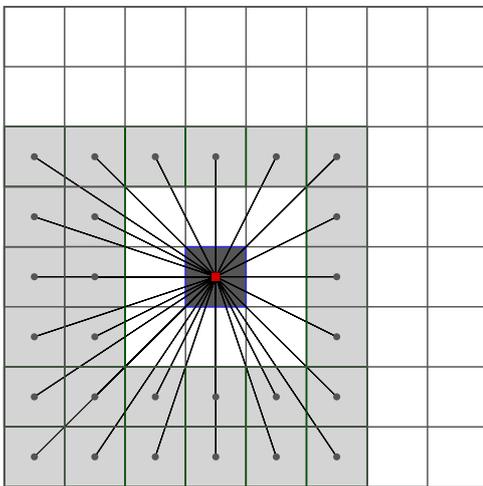
Figure 3.7: Upward pass schematics.



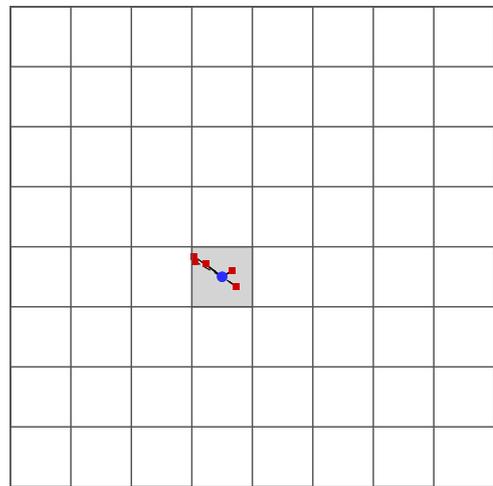
(a) Multipole-to-local at level 2.



(b) Local-to-local from level 2 to 3.



(c) Multipole-to-local at level 3.



(d) Local-to-particle at finest level.

Figure 3.8: Downward pass schematics.

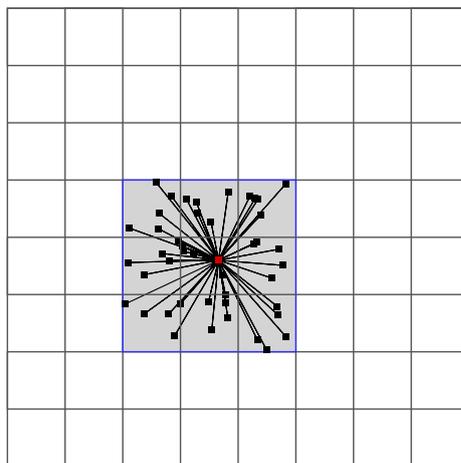
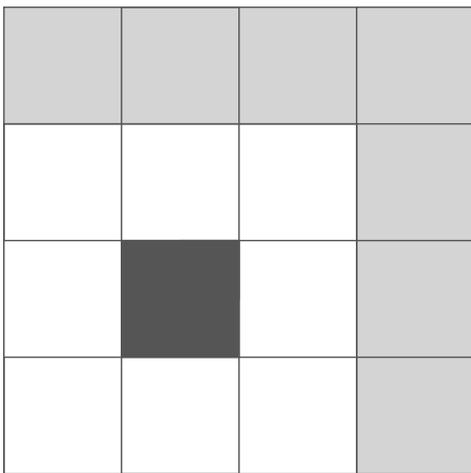
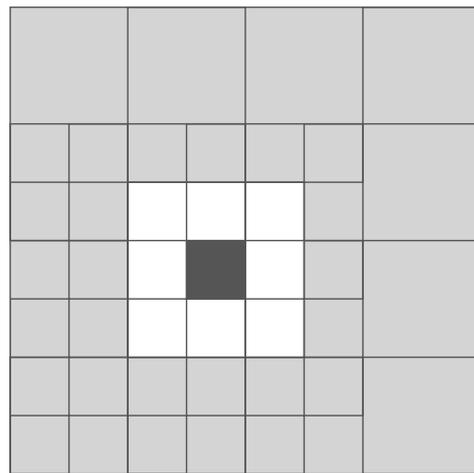


Figure 3.9: Near-field calculations schematics.

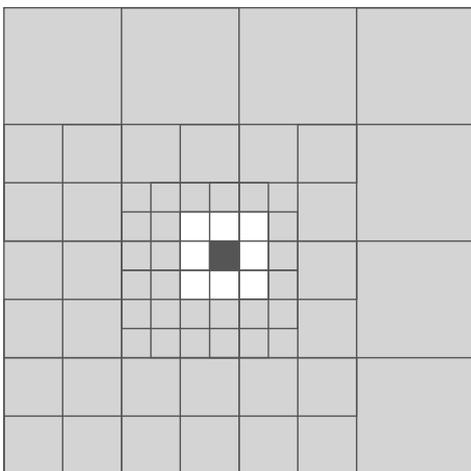
An overview of the far-field interactions computed from well-separated boxes at particular levels is shown in Fig. 3.10 for a generic box, indicated in dark gray. The operations must be performed for all boxes in the finest level. Figure 3.10(a) presents the far-field interactions for the larger boxes indicated by the light gray color. Increasing the refinement level, the far-field interactions occur for smaller clusters that could not be solved previously because of the separation criterion. Figure 3.10(b) and (c) shows the far-field interactions computed for levels 2, 3 and 4 from the light gray boxes to the dark one. This process is repeated until the clusters are small enough, such as in Fig. 3.10(d). These operations characterize a multi-level strategy to efficiently divide-and-conquer. This way, in Fig. 3.10(d), all light gray boxes already interacted with the dark gray box via FMM, either directly at the same level or indirectly at coarser levels. Then, only for the finest level  $L$ , the white boxes (neighbors) interact directly with the dark gray box solving the Green's function due to the separation criterion of the FMM.



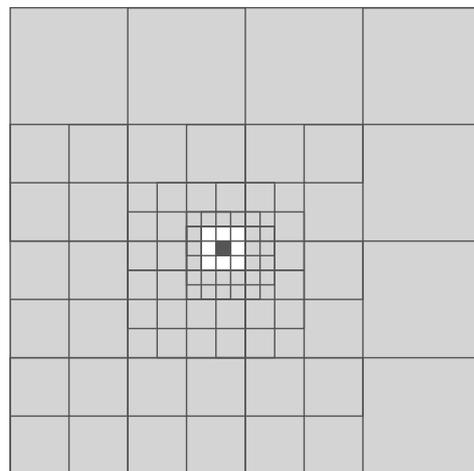
(a) Level 2.



(b) Level 2 + level 3.



(c) Level 2 + level 3 + level 4.



(d) Level 2 + level 3 + level 4 + level 5.

Figure 3.10: Multi-level interactions in the far-field (in light gray).

Resuming the algorithm: with exception of the finest level neighbors, all boxes interact via the FMM using the M2L operations. This interaction happens directly if they are in the same level or indirectly by the ancestors from a combination of M2M, M2L and L2L operations. When performed hierarchically, this clustering methodology is the main mechanism to reduce the far-field cost, while the near-field cost is reduced using smaller clusters. This combination reduces the computational cost from  $O(N^2)$  towards  $O(N)$ .

### 3.1.3 Computational cost

The computational cost of the main steps, illustrated from Fig. 3.6 – 3.9, in the FMM are given below.

- Step 1, the mapping depends on the number of particles ( $N$ ) and refinement level ( $L$ ); total cost proportional to  $(N) \times (L)$
- Step 2, the cost of clustering the elements into multipole expansion in the finest level depends linearly on the number of particles ( $N$ ) and the series precision ( $p$ ); total cost proportional to  $(N) \times (p)$
- Step 3, the multipole-to-multipole is performed from child to parent and its cost is proportional to the square of the precision ( $p^2$ ) and it is linear with the number of boxes ( $4^L$ ); total cost proportional to  $(p^2) \times (4^L)$
- Step 4, the multipole-to-local operation cost depends on the square of the precision ( $p^2$ ) and it is linear with the number of boxes ( $4^L$ ). Also, this operation is performed for all 27 boxes in interaction list; total cost proportional to  $27 \times (p^2) \times (4^L)$
- Step 5, the local-to-local operation is performed from parent to child and its cost is proportional to the square of the precision ( $p^2$ ) and it is linear with the number of boxes ( $4^L$ ); total cost proportional to  $(p^2) \times (4^L)$
- Step 6, the local-to-particle interactions depend linearly on the precision ( $p$ ) and on number of particles, ( $N$ ); total cost proportional to  $(N) \times (p)$
- Step 7, the direct summation cost, among the particles within the 8 neighbor boxes and the box itself, is linearly proportional to the number of boxes ( $4^L$ ) and to the square of the mean local number of elements  $(N/4^L)^2$ ; total cost proportional to  $9 \times (N^2) \times (1/4^L)$

Combining the individual costs of these steps based on the dependence of the number of particles  $N$ , the overall complexity of the algorithm is given by the following expression:

$$\mathcal{A}_2 N^2 + \mathcal{A}_1 N + \mathcal{A}_0 , \quad (3.3)$$

where the quadratic dependence is given by

$$\mathcal{A}_2 = f\left(\frac{1}{4^L}\right) . \quad (3.4)$$

The linear parameter depends on

$$\mathcal{A}_1 = f(L, p) , \quad (3.5)$$

while a fixed cost depends on

$$\mathcal{A}_0 = f(4^L, p^2) . \quad (3.6)$$

Equation 3.3 shows a cost reduction of the algorithm towards  $O(N)$  for small  $\mathcal{A}_2$ . However, the remaining dependency on  $(N^2)$  for FMM-DVM results in an intermediate computational cost:

$$O(N) < \text{FMM} < O(N^2) .$$

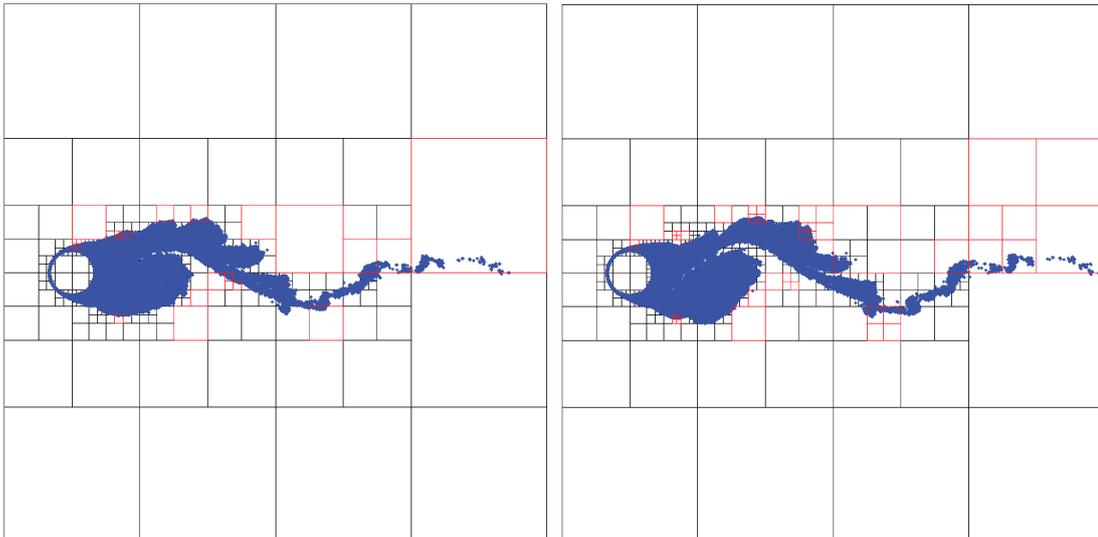
The  $\mathcal{A}_2$  parameter depends on the spatial distribution of elements, while  $\mathcal{A}_1$  and  $\mathcal{A}_0$  depend also on precision, implementation details and spatial arrangement of the elements. This way, a good estimate for the total cost is not so easily evaluated. Also, all parameters depend on the refinement level through the number of boxes. A hierarchical, multi-level approach can be applied to exploit more of the divide-and-conquer strategies and further accelerate the problem by dividing larger clusters into smaller ones.

Increasing the maximum refinement level,  $L$ , it is possible to reduce  $\mathcal{A}_2$  and, therefore, the quadratic dependence of the method. However, one would increase both the fixed cost of clustering  $\mathcal{A}_0$  and the linear coefficient  $\mathcal{A}_1$ . A rough estimative is that for a unitary increment in the level of refinement, the quadratic coefficient reduces by nearly one fourth while both linear and fixed costs increase four times. The memory costs also increases nearly four times for each level of refinement.

This logic indicates that, for a given number  $N$  of elements, an excellent level exists when both far-field and near-field costs are similar. For small  $N$ , the direct computation of the Biot-Savart will be faster due to the absence of the fixed cost. On the other hand, simulations performed with the FMM, for a large number of elements, will benefit from the efficiency of method. An assessment of the computational costs of the FMM is presented in section 4.3.4.

## 3.2 Assessment of the FMM-DVM

The objective of this work is to apply the fast multipole method to accelerate the solution of the discrete vortex method. The latter is applied for the simulation of unsteady flows where the vortex elements move every time step. This non-stationary problem creates new challenges that do not exist in static calculations using the FMM. The motion of the particles enables them to cross the boundaries of boxes in the FMM, and regions that are highly populated with vortices at a time instant can be empty a few steps later and vice-versa. The results of a simulation of the flow around a cylinder exemplifies these problems. It is shown in Fig. 3.11 red boxes that have been either further refined or became empty after a few time-step of computation, Fig. 3.11(a) and (b). If one is working with the global refinement approach, these problems are less critical since the partitioning is performed for the entire FMM domain and it is also independent of the location of particles. However, for the adaptive refinement approach, particles can migrate to regions with nonexistent boxes so the preprocessing step has to be performed again.



(a) Solution at  $t^* = 12.25$ .

(b) Solution at  $t^* = 13.00$ .

Figure 3.11: Motion of vortex particles and changes in adaptive refinement.

### 3.2.1 Preprocessing steps

As explained in the preprocessing steps in section 3.1.1, the creation of boxes in the global refinement approach is faster than the adaptive one since it does not depend on the number of particles. It is only dependent on the number of boxes and maximum refinement level so it is faster than the adaptive case. Also, both interaction and neighbor

lists are independent of the domain size, spatial displacement or number of elements, so they do not need to be performed every time step. On the other hand, for the adaptive approach, changes in the location of vortex elements affect the refinement of boxes and all five interaction lists may have to be recalculated every time step. The creation of the lists has to be performed with a convolution in all boxes and, hence, the preprocessing computational cost is proportional to the square of the number of boxes. Also, in order to refine boxes, it is necessary to know how many elements they contain until the criterion of maximum number of elements per box is satisfied. With that, the preprocessing cost is also proportional to the number of elements. Although the cost of far-field interactions *per se* is faster for this approach, the increased cost in preprocessing adds a penalty.

Although the far-field interactions performed using the global refinement are more costly than those for the adaptive approach, for dynamic cases with moving particles, such as in the DVM, it is possible to exploit the benefit that the first approach is less sensitive to the motion of particles and, thus, one can avoid constant preprocessing to save computational time. In this work, we propose to use a global refinement approach with modifications to further exploit the lower computational cost of its preprocessing step as well as the weaker dependence of the elements' spatial distribution. Details of the computational algorithm developed in this work are given in the Appendix A, and an illustrative example of the modifications proposed are described in section 3.2.3, where the flow around a cylinder is investigated.

In the current implementation, the preprocessing step is only performed in two situations: whenever a particle leaves the FMM domain or when the density of particles becomes too high so the near-field computations become too expensive. To solve the issue from the first situation, the FMM domain size  $S_0$  becomes slightly larger than the wake in order to involve all elements. Hence, obligatorily all boxes in all levels also grow so the new position of their center must be calculated. Since the number of boxes per line in the refinement level  $\ell$  is equal to  $2^\ell$ , the size  $S_\ell$  of the boxes at level  $\ell$  is

$$S_\ell = \frac{S_0}{2^\ell}. \quad (3.7)$$

One advantage of this scheme is that the refinement is not modified, so the neighbor and interaction lists do not change and this step of the preprocessing is very cheap.

The second criteria to perform preprocessing in the current implementation is when the density of vortex elements increases and, therefore, it is necessary to further refine the FMM domain as explained in section 3.1.3. This way, it is required not only to create boxes in the new finest level but also to recompute the interaction and neighbor lists for all boxes, being them empty or no. The allocation of these lists for all boxes avoids its constant computation. However, since they must be always allocated for all boxes in all levels of

refinement, efficient ways to avoid excessive memory costs are necessary. In the DVM, vorticity is present along regions containing boundary layers and aerodynamic wakes which may extend downstream for several reference characteristic lengths, creating regions with high density of elements. This way, the FMM domain requires higher refinement levels to reduce the near-field cost. This problem creates thousands of empty boxes which increase both memory and far-field costs.

Following the main idea of the adaptive algorithm, empty boxes are ignored to reduce computational cost in the FMM steps 2 through 7. The list of non-empty boxes are determined together with the mapping of the particles (step 1) and they are computed at every sub-step of any time marching method, so efficient procedures are required. A fast way proposed in this work to map the particles is based on the genealogical tree. Recursively comparing the  $x$  and  $y$  coordinates of any particle with respect to the center of a box at level  $\ell - 1$ , it is possible to determine the level  $\ell$  box which contains the particle. This procedure, based on quadrants, repeats until the finest level is achieved, as shown in Fig. 3.12. An auxiliary variable counts how many particles are inside the boxes, leading to the list of non-empty boxes.

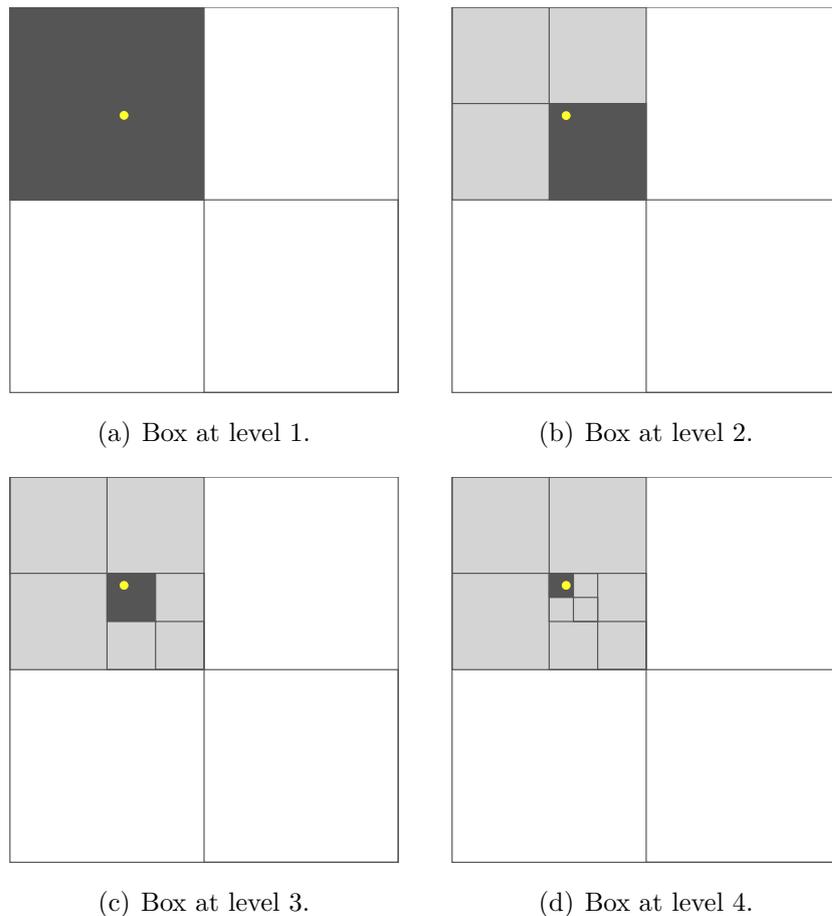
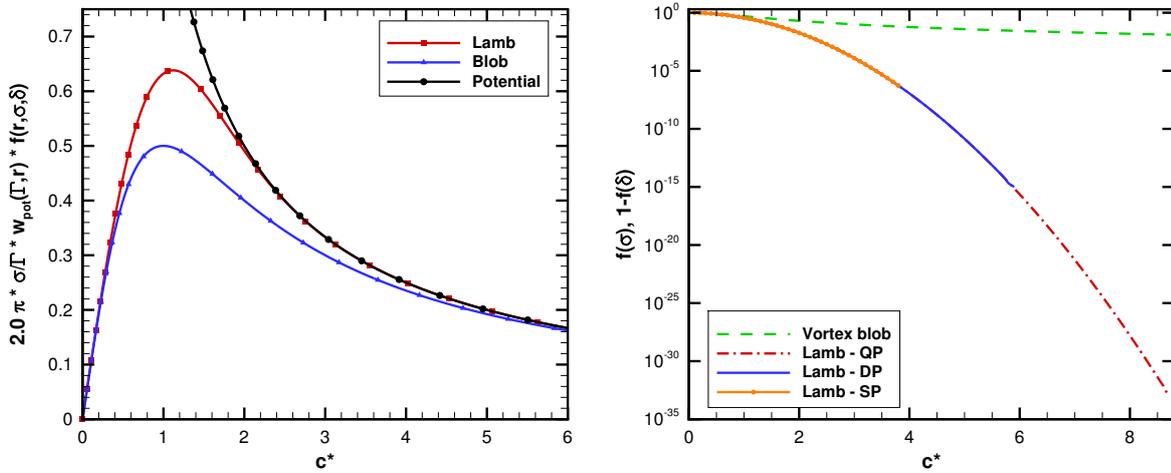


Figure 3.12: Efficient way for mapping the location of vortex elements.

### 3.2.2 Desingularization models

The kernel of the original FMM (Greengard and Rokhlin, 1987) is given by the logarithmic function (potential of a singular vortex) and/or its derivatives (velocity field induced by a potential vortex). However, both Lamb-Oseen and blob vortex models have terms that affect the velocity in the near-field, through diffusion, and converge to a potential vortex solution in the far-field, which is the region of interest in the FMM, as explained by Nishimura (2002). In order to choose a suitable model, an initial study of *how far is the far-field* is necessary for both Lamb-Oseen and blob models. Self-similar profiles of induced velocity for the two models are shown in Fig 2.4 and Fig. 3.13(a). Increasing  $r \rightarrow \infty$ , both converge to the potential vortex, as shown in Fig. 2.5 and Fig. 3.13(b). However, in the FMM, the regions  $r$  of interest for calculations are given by a few box lengths. The convergence of the smoothing parameter in Eqs. 2.48 and 2.50 indicates that the exponential term decays to zero for a box about 9 times the size of the vortex core, while the vortex blob model by Krasny (1986b) slowly converges algebraically to the potential vortex.



(a) Velocity profiles.

(b) Smoothing parameter.

Figure 3.13: Comparison between the Lamb-Oseen and vortex blob models.

With that in mind, we use the Lamb-Oseen vortex for all simulations and impose that the smallest box in the FMM is sufficiently larger than the largest core in the domain, based on the machine precision as shown in Fig. 3.13(b), for single, double and quadruple precision. Therefore, the FMM does not compromise neither the physics nor the numerical aspects of the simulation. In other words, the refinement level is now limited not only by the computational time for a single time step but also for the largest viscous core.

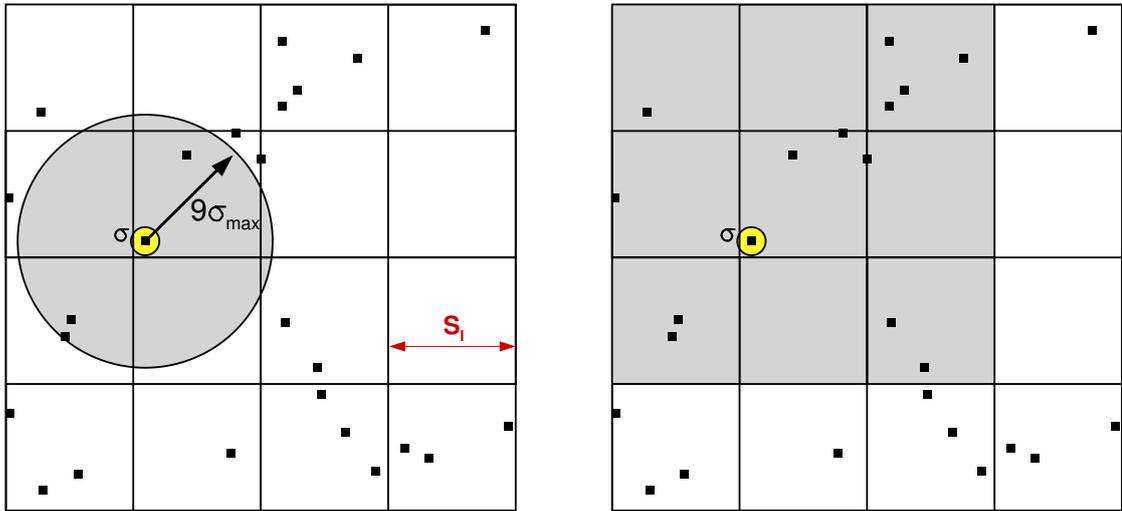
This criterion is illustrated in Fig. 3.14(a) with a generalization for the most restrictive computation using quadruple precision. It is imposed that the size of the smallest box,

$S_\ell$ , is at least 9 times the size of the largest core in the domain

$$S_\ell > 9\sigma_{max} . \quad (3.8)$$

For practical purposes, the direct calculations performed for particles in neighbor boxes (near-field) are accounted using the Lamb-Oseen vortex, Fig. 3.14(b), while for far-field FMM interactions both models (potential and Lamb-Oseen) have the same solution.

If this condition is not satisfied, there are spurious perturbations proportional to the smaller FMM box size, as shown in Fig. 3.15, where the over-refined level  $L = 4$  is used and the size of the box is  $S_L = 0.0625$ . The viscous core is  $\sigma = 0.05$ , and, although it is smaller than the box this is not sufficient to avoid spurious disturbances.



(a) Influence region of the smoothing parameter.

(b) Interaction region using Lamb vortex.

Figure 3.14: Details on the use of Lamb-Oseen vortex model applied to the FMM.

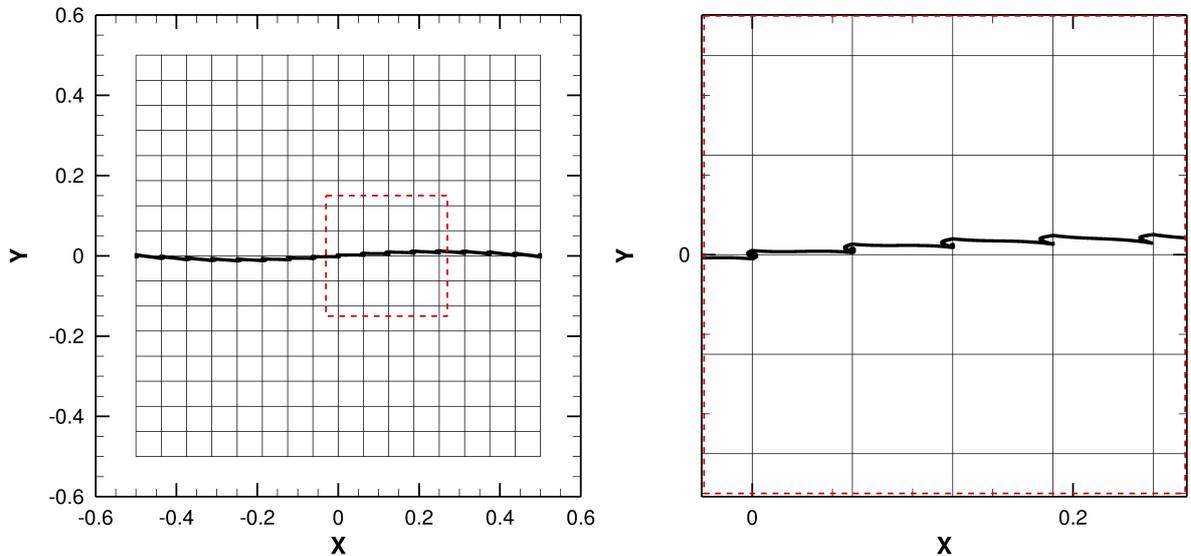


Figure 3.15: Spurious perturbations appearing at refinement level 4 and  $\sigma = 0.05$ .

### 3.2.3 Example - flow around a circular cylinder

In this section, an example of the application of the fast multipole method coupled to the discrete vortex method is presented. Here, the FMM is used to accelerate the solution of the flow around a cylinder. This simulation was previously performed by Bimbato *et al* (2009) who employed a first-order RK1 time marching scheme for the convection of vortex particles. These authors used a time step  $\Delta t = 0.05$  and the Random Walk Method, RWM, to solve the diffusion process for a Reynolds number of  $1 \times 10^5$ . The body surface was discretized with 300 flat panels with source distribution. Furthermore, at every time step, 300 new vortex particles are placed at a fixed distance  $\sigma = 0.001$  normal to the cylinder's wall to enforce the no-slip boundary condition. Therefore, the computational cost of the present simulation grows in time with the creation of new particles. A cylinder with diameter  $D = 1$  is placed at the origin of the Cartesian coordinate system and the square FMM domain, at level 0, with initial size  $S_0 = 3.0$ , is placed from  $-1 \leq x \leq S_0 - 1$  and  $-S_0/2 \leq y \leq +S_0/2$ . The imposed initial maximum refinement is  $L = 4$ .

As explained before, when the cost of near-field calculations becomes 3 times that of the far-field, a new refinement level is added to reduce the cost of near-field by sacrificing that of far-field. The choice of this parameter "3" is determined by numerical inspection and depends mainly on the proportion of empty boxes after the refinement. Only the subroutines responsible for the refinement are called in the preprocessing, without changing the level 0 box size, as shown in Fig. 3.16. Also, whenever a vortex leaves the domain, some subroutines of the preprocessing are recomputed to increase the FMM domain size without changing the refinement level and the interaction list. This procedure is shown in Fig. 3.17, for refinement level 5. Also, the near-field cost increases when more vortex particles are present in the domain, as discussed in section 3.1.3.

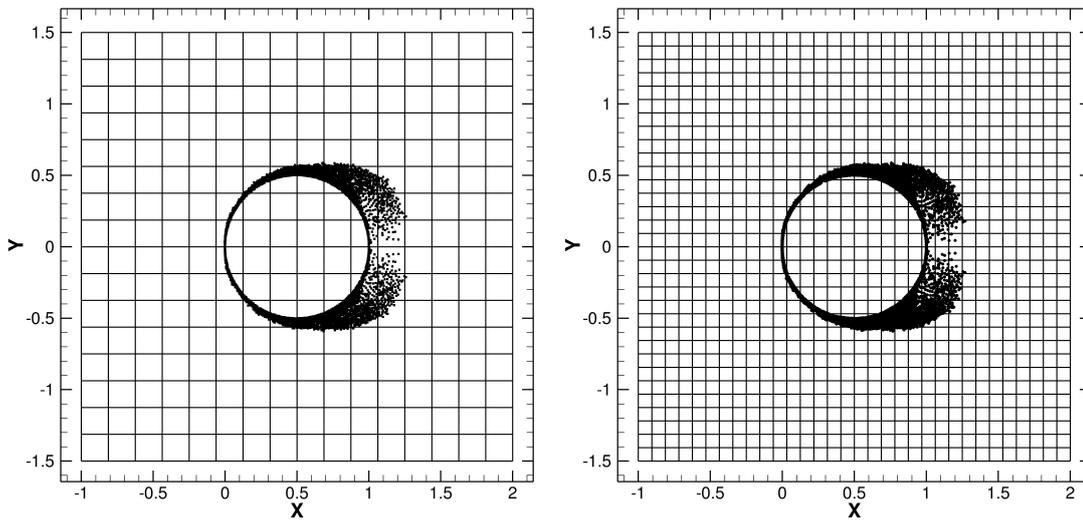
The computational time of the current FMM implementation is shown in Fig. 3.18. For 300k particles, the convolution time with the FMM is about 8 seconds, while direct summation with the Biot-Savart law has a cost of 1000 seconds. The vertical dashed lines indicate when a new refinement level is added, ranging from 4 to 9 levels during the current analysis. One can see in this figure that the cost of the near-field calculations grows faster than that of the far-field interactions. From this figure, it is also possible to notice that the FMM has an overall computational cost  $O(N^{1.3})$  while the direct evaluation of the Biot-Savart has a cost proportional to  $O(N^2)$ . The FMM was used only to accelerate the convolution among the  $N$  vortex elements. However, it is possible to use the FMM to accelerate the vector at the right side for vortex generation, Eq. 2.82, using the same upward pass for the convolution. The only modification necessary in the algorithm is to reevaluate the downward pass for the observers located at the center of each panel.

It is possible to see in Table 3.1 a good agreement between the FMM and Biot-Savart results compared to the experimental data from Lienhard (1966). Also, due to the RWM, a stochastic method which adds a random displacement of the vortices, the results from FMM computations differ slightly to the Biot-Savart direct.

Figure 3.19 shows the development of the cylinder wake where one can visualize a von-Karman vortex street. There, the blue dots are vortices with counter-clockwise circulation, while the reds are clockwise. In Fig. 3.20, one can see plots of unsteady lift and drag coefficients after the initial transient. Finally, Fig. 3.21 presents the spectrum of lift coefficient in the wake showing the tonal component characteristic of vortex shedding. From this figure, one can see the Strouhal number excited by the von-Karman vortex street.

Table 3.1: Comparison of aerodynamic coefficients for FMM-DVM, DVM and experimental data.

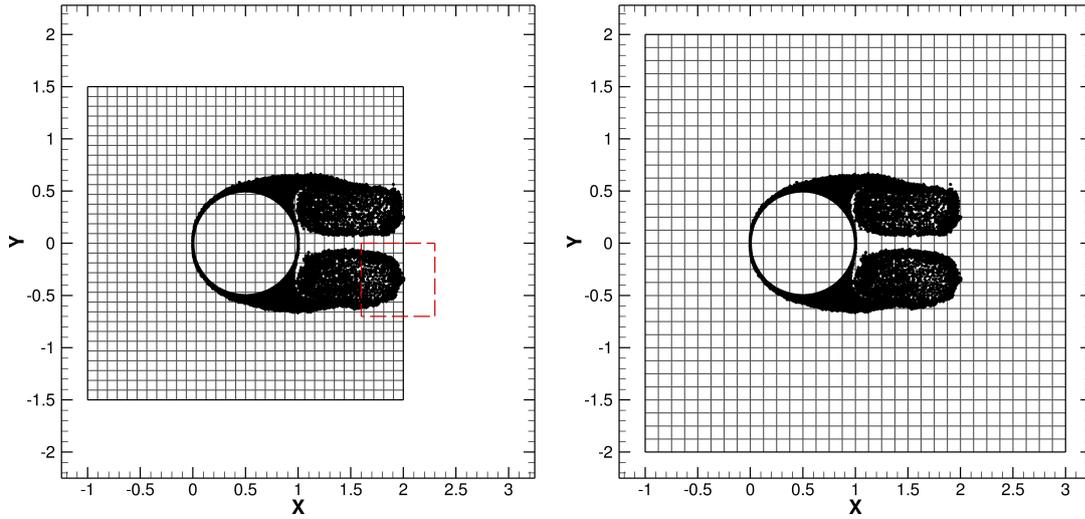
Parameter	FMM	Biot-Savart	Experimental
Mean lift	-0.025	-0.0101	-
Mean drag	1.1809	1.2157	1.1 – 1.2
Strouhal number	0.2214	0.2226	0.195 – 0.215



(a) Maximum level  $L = 4$  in step 20.

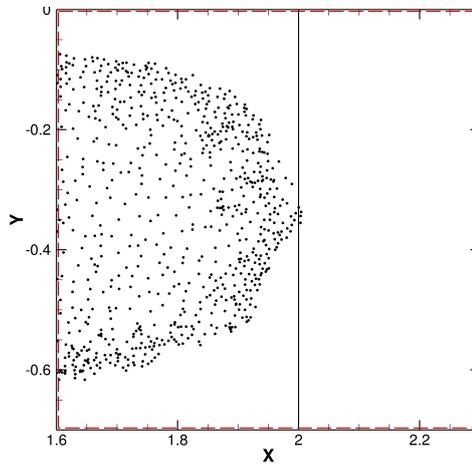
(b) Maximum level  $L = 5$  in step 21.

Figure 3.16: FMM domain refinement during step 20 to step 21.



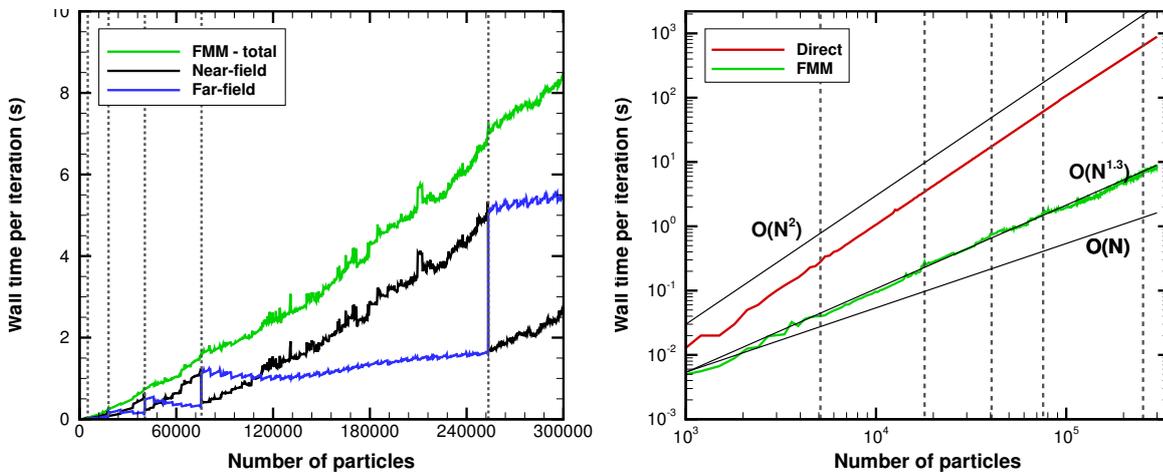
(a) Domain with length  $S = 3$ .

(b) Domain with length  $S = 4$ .



(c) Detail of vortex leaving the domain.

Figure 3.17: FMM domain growth due to the escape of particles.



(a) FMM convolution time.

(b) Convolution time for both methods.

Figure 3.18: Computational time for the flow around a circular cylinder.

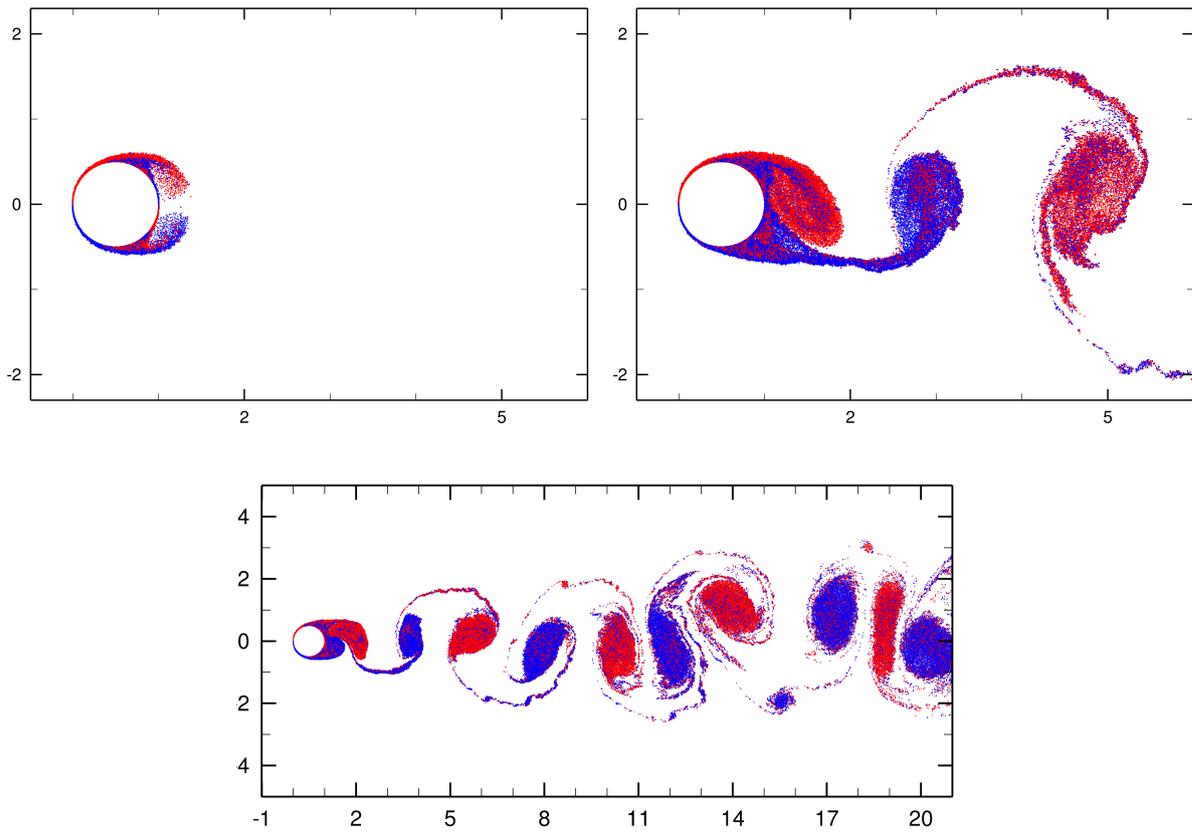


Figure 3.19: Development of circular cylinder wake.

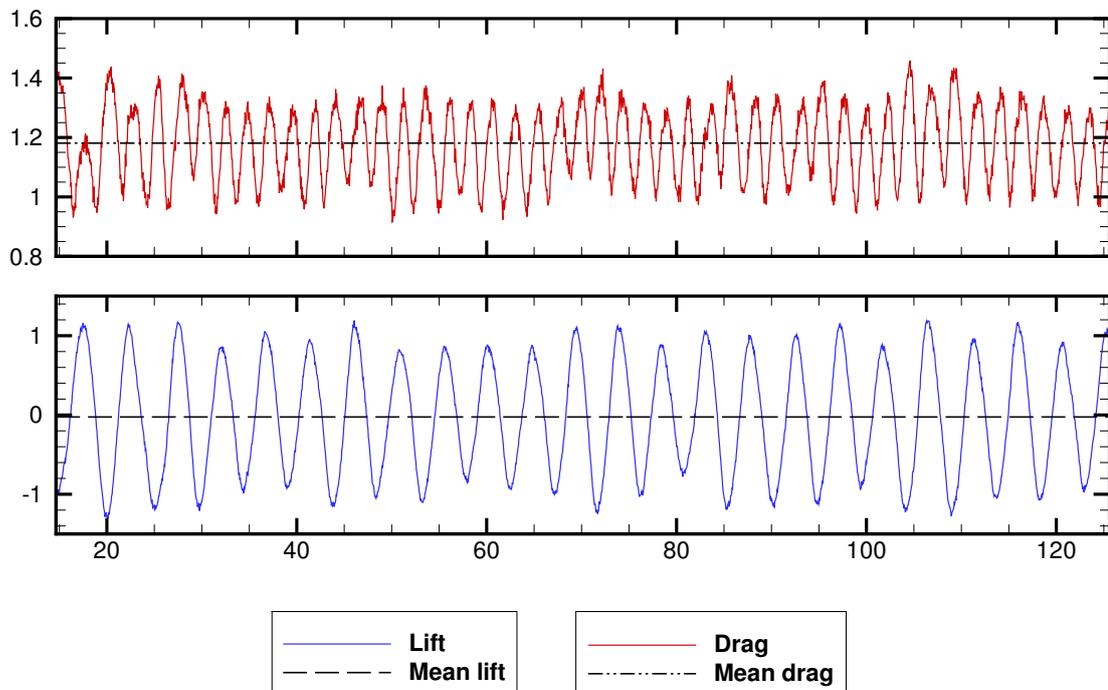


Figure 3.20: Aerodynamic coefficients for the circular cylinder.

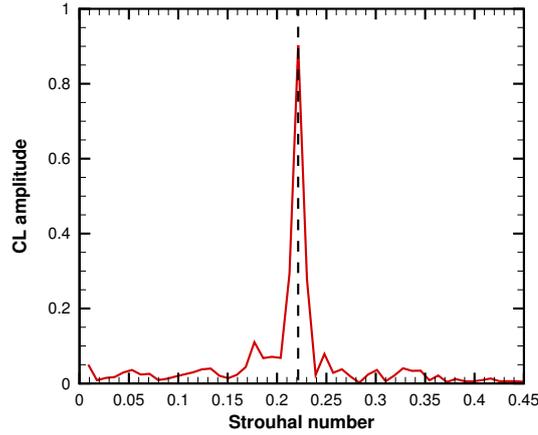


Figure 3.21: Lift spectrum showing the Strouhal number for the cylinder vortex shedding.

### 3.3 Mathematical formulation

The main equations used in the fast multipole method implementation are shown in the present section. They can be also found in the work of Greengard and Rokhlin (1987) with slight changes in nomenclature. They are reproduced here for completeness and, in some cases, they are modified for implementation in the DVM. For more details about the formulations, one should see the original reference.

The potential  $\phi$  at a distant observer  $z_o$  from a source with unitary strength placed at  $z_s$  is given by the Green's function

$$\phi_s(z_o) = G(z_s, z_o) = \log(z_o - z_s) , \quad (3.9)$$

for the observer and source position, respectively,

$$z_o = x_o + iy_o \quad (3.10)$$

and

$$z_s = x_s + iy_s , \quad (3.11)$$

where

$$\left| \frac{z_s}{z_o} \right| < 1 . \quad (3.12)$$

In the equations above,  $x$  and  $y$  are Cartesian coordinates and the indices  $o$  and  $s$  represent the observer and source, respectively. The Green's function is further manipulated to

$$G(z_s, z_o) = \log(z_o - z_s) = \log\left(1 - \frac{z_s}{z_o}\right) + \log(z_o) . \quad (3.13)$$

The first term in Eq. 3.13 can be further expanded in a Taylor series as

$$\log(1-x) = (-1) \left( \frac{x^1}{1} + \frac{x^2}{2} + \frac{x^3}{3} + \dots \right) = (-1) \sum_{k=1}^{\infty} \frac{1}{k} \frac{z_s^k}{z_o^k}, \quad (3.14)$$

so one can isolate the source terms from the observers, as required by the FMM algorithm.

The error bound of the FMM according to the series truncation and with the separation criterion of one box is given by

$$\epsilon \sim \left( \frac{1}{2} \right)^p, \quad (3.15)$$

and the number of terms used in the series for an specified precision is

$$p = -\log_2(\epsilon). \quad (3.16)$$

The separation criterion could be larger than one box, increasing the method precision. However, the interaction among close boxes as well in the well-separated boxes would be larger, increasing the computational cost. On the other hand, it is possible to reduce the number  $p$  of terms in the series. However, it is still not enough to keep the same cost as that from the one box separation criterion.

### 3.3.1 Far-field

#### 3.3.1.1 Upward pass

In the FMM, mathematical relations are used to cluster scattered sets of individual particles into a larger equivalent source. Let us consider a set of  $N_b$  sources with intensity  $\Gamma_s$ , for a clockwise vortex given by  $\Gamma > 0$ , inside a box  $b$  centered at  $z_b$  at the finest level  $L$ . The sources are clustered in the particle-to-multipole step given by Eqs. 3.17 – 3.19. Using the series given by Eq. 3.14 and truncating it after  $p$  terms, one can write the intensities of source clusters using the following multipole expansions

$$\mathcal{M}_0(b, L) = \sum_{s=1}^{N_b} \Gamma_s, \quad (3.17)$$

$$\mathcal{M}_k(b, L) = - \sum_{s=1}^{N_b} \frac{\Gamma_s z^k}{k}. \quad (3.18)$$

The terms  $\mathcal{M}_k(b, L)$ , with  $k = 0 \dots p$ , represent the intensity of the cluster of sources inside box  $b$  at level of refinement  $L$  and  $z$  is the complex distance from the source  $s$  to the box  $b$  center given by

$$z = z_s - z_b. \quad (3.19)$$

The same idea of clustering scattered sources into an even larger source can be used in a recursive step. The clusters of sources within children boxes  $b_c$  at level  $\ell + 1$ , for  $0 < \ell < L - 1$ , centered at  $z_{b_c}$ , can be re-grouped in larger clusters at their parent box  $B$ , which is centered at  $z_B$  at level  $\ell$ . This operation is called multipole-to-multipole expansion and it is given by Eqs. 3.20 – 3.22. One can see this step as an upward pass in the genealogical tree of the multi-level FMM: it is performed from the finest level  $L$  up to level 2 for free-domain problems or up to level 0 for periodical problems.

$$\mathcal{M}_0(B, \ell) = \sum_{c=1}^4 \mathcal{M}_0(b_c, \ell + 1) , \quad (3.20)$$

and, for  $k = 1$  to  $p$ :

$$\mathcal{M}_k(B, \ell) = \sum_{c=1}^4 \left\{ \sum_{m=1}^k \left[ \binom{k-1}{m-1} \mathcal{M}_m(b_c, \ell + 1) \cdot z^{k-m} \right] - \mathcal{M}_0(b_c, \ell + 1) \frac{z^k}{k} \right\} . \quad (3.21)$$

where the complex distance from the child box  $b_c$  to its parent box  $B$  is given by

$$z = z_{b_c} - z_B . \quad (3.22)$$

Also in Eq. 3.21, the binomial coefficient is computed using factorial as:

$$\binom{k-1}{m-1} = \frac{(k-1)!}{(m-1)!(k-m)!} . \quad (3.23)$$

However, factorials may be large numbers and the operation shown in Eq. 3.23 may lead to loss of precision due to machine truncation during computations. A better way to evaluate the binomial coefficient, if  $m < k - m$ , is

$$\binom{k-1}{m-1} = \frac{(k-1)(k-2)\dots(k-(m-1))(k-m)!}{(m-1)!(k-m)!} = \prod_{j=1}^{m-1} \frac{k-j}{m-j} . \quad (3.24)$$

On the other hand, if  $m > k - m$ , the coefficient is calculated as

$$\binom{k-1}{m-1} = \frac{(k-1)(k-2)\dots(k-(k-m))(m-1)!}{(k-m)!(m-1)!} = \prod_{j=1}^{k-m} \frac{k-j}{j} . \quad (3.25)$$

The procedure for upward pass operations is illustrated in Fig. 3.7(c) - P2M at level  $L$  and Fig. 3.7(e) - M2M at level  $L-1$ .

### 3.3.1.2 Downward pass

Once multipole expansions are computed representing clusters of sources, it is possible to calculate their far-field interactions on clusters of well-separated observers at the same level, indicated by the interaction list. This operation is called multipole-to-local and is given by Eqs. 3.26 – 3.28. The cluster at box  $b_S$  is one of the 27 in the interaction list of box  $b$ . The multipole-to-local step is employed downward the tree and it begins in the coarsest level (0 or 2 depending on the boundary conditions) until the finest refinement level  $L$ . In the equations below, the terms  $\mathcal{L}_k(b, \ell)$ , with  $k = 0$  to  $p$ , represent the effects of far-field clusters of sources  $b_S$  at a local box  $b$  at level  $\ell$ .

$$\mathcal{L}_0^{(M2L)}(b, \ell) = \sum_{S=1}^{27} \left[ \mathcal{M}_0(b_S, \ell) \cdot \log(-z) + \sum_{k=1}^p \left( \mathcal{M}_k(b_S, \ell) \frac{(-1)^k}{z^k} \right) \right], \quad (3.26)$$

where the complex distance from the source cluster  $b_S$  to the observer,  $b$ , is

$$z = z_{b_S} - z_b. \quad (3.27)$$

For  $m = 1$  to  $p$ :

$$\mathcal{L}_m^{(M2L)}(b, \ell) = \sum_{S=1}^{27} \left\{ -\frac{\mathcal{M}_0(b_S, \ell)}{m \cdot z^m} + \frac{1}{z^m} \sum_{k=1}^p \left[ \binom{m+k-1}{k-1} \mathcal{M}_k(b_S, \ell) \frac{(-1)^k}{z^k} \right] \right\}. \quad (3.28)$$

The transference of the local effects in a large box at level  $\ell - 1$  to a smaller box  $b$  at level  $\ell$  through its parent  $B$  uses the local-to-local operation, given by Eq. 3.29, for  $m = 0$  to  $p$ :

$$\mathcal{L}_m^{(L2L)}(b, \ell) = \sum_{k=m}^p \binom{k}{m} \cdot \mathcal{L}_k(B, \ell - 1) \cdot (-z)^{k-m}, \quad (3.29)$$

with the complex distance from the parent box  $B$  to its child box  $b$  given by

$$z = z_B - z_b. \quad (3.30)$$

The calculations are performed for a parent box  $B$  from the coarsest level until the penultimate level  $L - 1$ , since the boxes at the finest level are childless.

The multipole-to-local (M2L) operations, Eq. 3.26 – 3.28, are performed to account for the interactions of far-field clusters of sources contained in boxes at the same level of refinement. Then, this information is translated down to finer levels of refinement using local-to-local (L2L) operation, Eq. 3.29. The M2L and L2L steps are coupled in order to add the contribution of clusters at coarser levels to those at the same level. In other words, the influence of far-field clusters of sources contained in larger boxes at coarser levels to finer boxes is accounted for in smaller boxes indirectly using its ancestors. These

two operations are recursively repeated from the coarser levels up to the finer level, so the local expansion  $\mathcal{L}_m(b, \ell)$  for box  $b$  at level  $\ell$  is computed by

$$\mathcal{L}_m(b, \ell) = \mathcal{L}_m^{(L2L)}(b, \ell) + \mathcal{L}_m^{(M2L)}(b, \ell) . \quad (3.31)$$

This procedure is illustrated in Fig. 3.8(a) - M2L at level  $\ell - 1$ , Fig. 3.8(b) - L2L from parent box to children, and Fig. 3.8(c) - M2L at level  $\ell$ .

After all far-field interactions are accounted for in  $\mathcal{L}_m$ , until the finest level  $L$ , the translation from local expansion to observer particle  $o$  is performed. This particle pertains to a cluster within box  $b$  and, at this box, the local-to-particle operation, Eqs. 3.32 – 3.33, is given by

$$\phi(z_o) = \sum_{m=0}^{\infty} \mathcal{L}_m(b, L) \cdot z^m . \quad (3.32)$$

In the equation above, the complex distance from the particle  $z_o$  to the box  $b$  center is given by

$$z = z_o - z_b . \quad (3.33)$$

One should remind that, in the DVM, it is necessary to evaluate the velocity field generated by the sources, which can be computed from the derivative of the potential. This leads to the complex velocity  $w$  at the observer  $z_o$ , written as

$$w(z_o) = \frac{\partial \phi(z_o)}{\partial z_o} = \frac{\partial [\log(z_o - z_s)]}{\partial z_o} = \frac{1}{z_o - z_s} . \quad (3.34)$$

Hence, the far-field influence is obtained from the analytical derivative of Eq. 3.32, which leads to:

$$w(z_o)^{(FF)} = \frac{\partial \phi(z_o)}{\partial z_o} = \sum_{m=0}^{\infty} \mathcal{L}_m(b, L) \cdot m \cdot z^{m-1} . \quad (3.35)$$

This equation can be further simplified because for  $m = 0$  the expression is null. This way, it is also not necessary to evaluate  $\mathcal{L}_0$  in Eq. 3.26 or Eq. 3.29. Only the terms  $\mathcal{L}_k$  for  $m > 0$  from the L2L and M2L operations in Eqs. 3.28 and 3.29, respectively, are required for the calculation of the complex velocity from the far-field. Furthermore, only the first  $p$  terms are evaluated. Hence,

$$w(z_o)^{(FF)} \approx \sum_{m=1}^p \mathcal{L}_m(b, L) \cdot m \cdot z^{m-1} . \quad (3.36)$$

### 3.3.2 Near-field induced velocity

After computing the velocity from far-field clusters, it is necessary to evaluate the near-field velocity from vortices at adjacent boxes which do not satisfy the separation criterion of the FMM. This calculation is performed using the Biot-Savart law for vortices  $n$ , which can be inside the 8 neighbor boxes of  $b_C$  or box  $b_C$  itself, to an observer  $z_o$  inside  $b_C$ . As all particles are close to each other, it is necessary to use the Lamb vortex in order to obtain a smooth velocity profile in the near-field given by

$$w(z_o)^{(NF)} = \sum_{j=1}^9 \sum_{n=1}^{N_j} \frac{\Gamma_n}{2\pi z} \left[ 1 - \exp\left(-\frac{|z|^2}{\sigma^2}\right) \right], \quad (3.37)$$

where the complex distance from source to observer is written as

$$z = z_o - z_n. \quad (3.38)$$

### 3.3.3 Total velocity

Combining both far and near-field calculations leads to the full velocity field in the original reference system as

$$w(z_o) = w(z_o)^{(NF)} + w(z_o)^{(FF)}, \quad (3.39)$$

where the kernel  $1/z$  is solved. For potential vortex, one should consider Eq. 2.44 where both the imaginary number  $i$  and the constant  $1/2\pi$  are present. Hence, the velocity components are given by

$$u(x_o, y_o) = -\text{Im}(w(z_o)) \frac{1}{2\pi}, \quad (3.40)$$

$$v(x_o, y_o) = -\text{Re}(w(z_o)) \frac{1}{2\pi}. \quad (3.41)$$

## 4 TEMPORAL EVOLUTION OF AIRCRAFT WAKE

Aerodynamic wakes of large aircraft and their induced velocity may cause serious hazards to smaller aircraft, affecting the take-off and landing operations in airports. Also, in-flight refueling can be a challenging task due to the rotational velocities imposed to the aircraft. Figure 4.1 shows the vortical structures formed by the aerodynamic wake of an aircraft, including the wing tip vortices. In Fig. 4.2, extracted from Ginevsky and Zhelannikov (2009), it is shown a typical airflow vertical velocity distribution in a cross-flow plane behind an aircraft wake.



Figure 4.1: Vortical structures formed by aerodynamic wake from wing tip and flap.

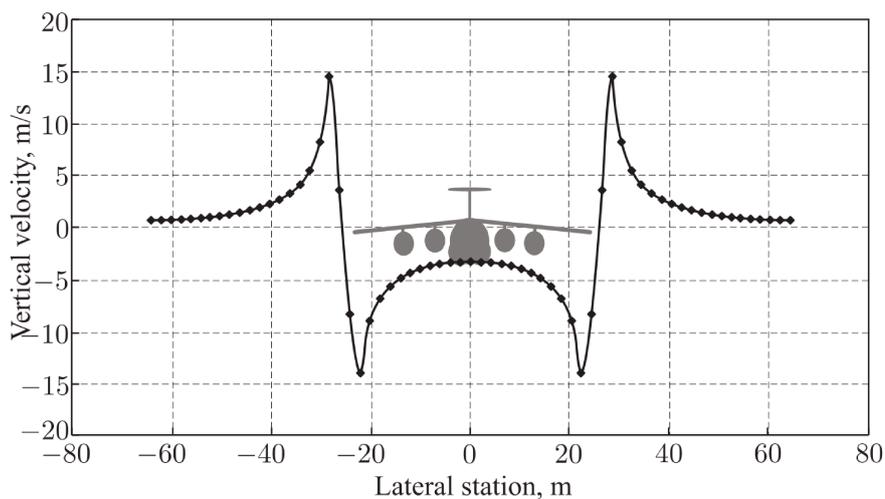


Figure 4.2: Typical airflow vertical velocity distribution in a cross-flow plane behind an aircraft (extracted from Ginevsky and Zhelannikov (2009)).

A simplified but representative model of an aircraft wake is given by Fig. 4.3, where the wake is discretized in vortex lines. A detailed explanation on the problem is provided by Smith (1986), Batchelor (2000) and Drela (2014). The DVM solves the temporal evolution of the vortex sheet in the Trefftz plane, as shown in Fig. 4.4, where a two-dimensional initial value problem models the vorticity shed by the wing. This figure also presents the pathlines for different vortex particles obtained by the DVM. As one can see, those close to the vortex core have higher angular frequencies with small amplitude, while particles further from the tip present lower frequencies but larger amplitude.

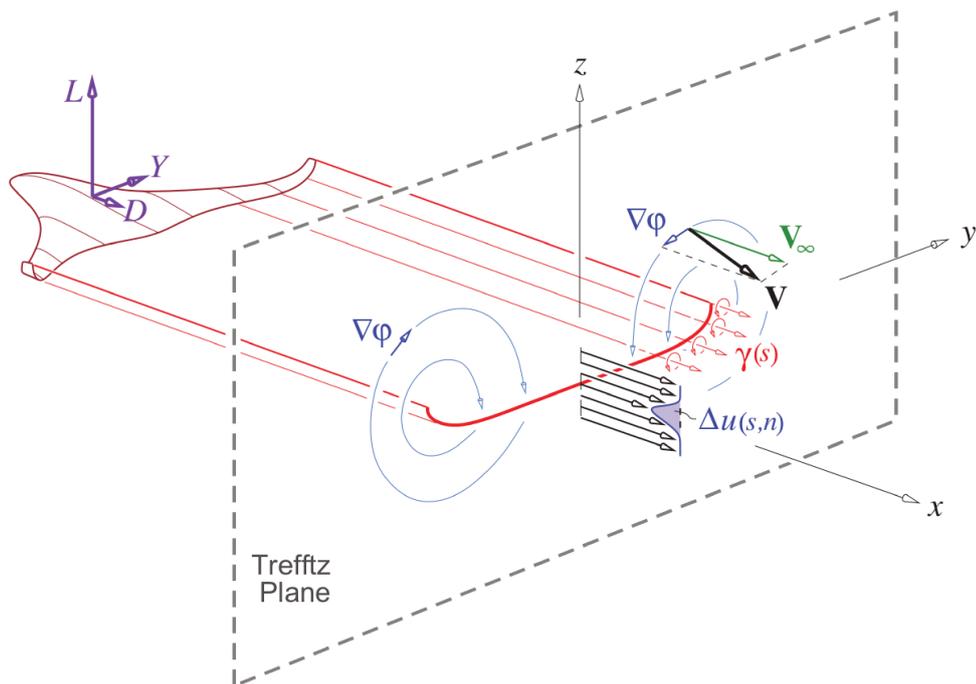


Figure 4.3: Model of aircraft wake in Trefftz plane (extracted from Drela (2014))

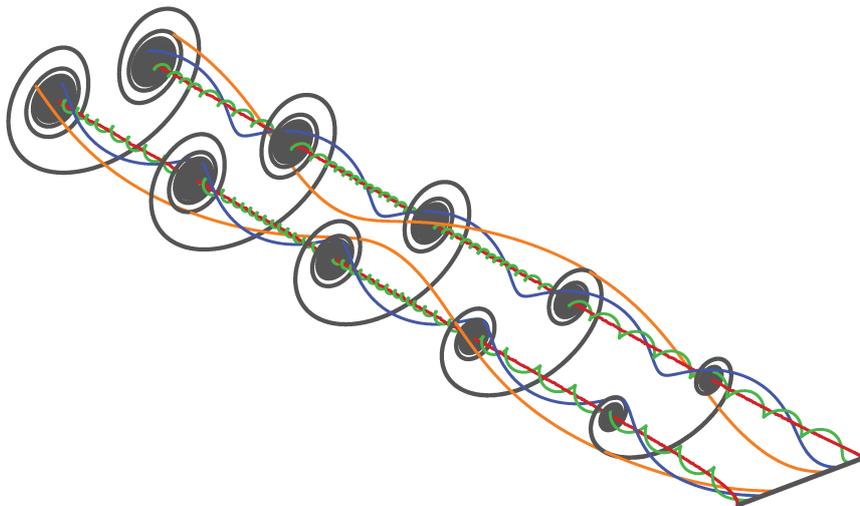


Figure 4.4: Temporal evolution of the Trefftz plane problem with pathlines for different particles given by colored lines.

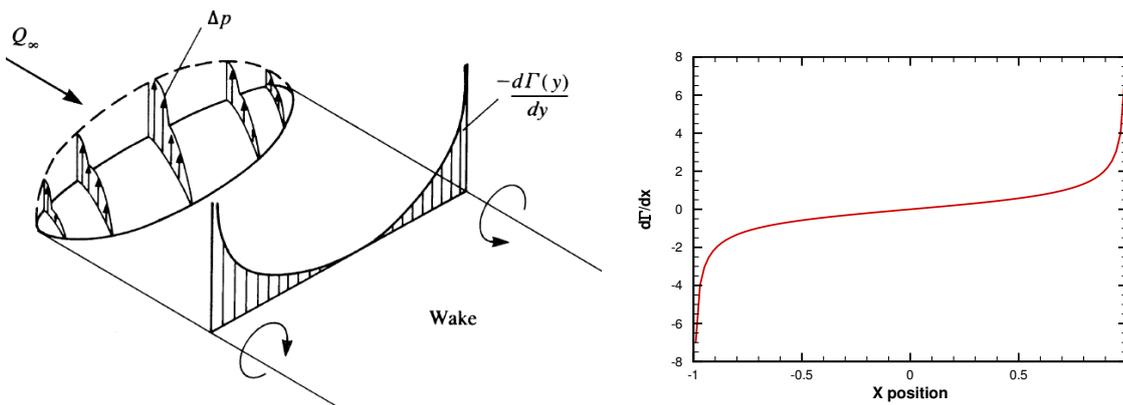
In this chapter, an assessment of the FMM accuracy is presented through a comparison with the direct solution of the Biot-Savart law. Both methods are applied to compute the temporal evolution of an aircraft wake. This is a very sensitive non-linear problem where accuracy is a critical factor, mainly towards a truly inviscid case with small desingularization value, *i.e.*,  $\sigma \rightarrow 0$ . The role of several parameters such as time marching schemes, time step restriction, number of particles in the wake discretization and machine precision is investigated mainly for numerical aspects of the problem. Finally, an investigation of viscous effects is presented.

## 4.1 Initial condition

Since the Trefftz plane is an initial value problem, the conditions of the wake have to be defined for the first time step and, then, its evolution is solved in time. The initial strength of the vortex-sheet is given by the derivative of the circulation along the wingspan, which is determined for an elliptically loaded wing, where the circulation varies smoothly along the wingspan, by Eq. 4.1 and shown in Fig. 4.5.

$$\frac{d\Gamma}{dx} = -\frac{x}{\sqrt{1-x^2}}. \quad (4.1)$$

In this case, the span is positioned from  $-1 \leq x \leq 1$ , with a singularity at  $|x| = 1$ . Other initial conditions can be used according to the lift distribution. For instance, and one can simulate the effects of deployed flaps or the presence of the fuselage, as shown by Krasny (1987).



(a) Illustrative change of lift on span - extracted from Katz and Plotkin (1991). (b) Calculated rate of variation of circulation along wingspan.

Figure 4.5: Rate of change in circulation along the span of an elliptic wing.

In order to set the initial condition for the temporal evolution, the wake is partitioned in  $N$  segments such that their boundaries are given by

$$x_{i+1}^n = x_i^n + \xi , \quad (4.2)$$

for  $i = 1$  to  $N + 1$ . The exponent  $n$  determines the concentration of segments, such that  $n = 1$  gives equal spacing, and  $n > 1$  concentrates them towards the tips. The parameter  $\xi$  is used to calculate the spacing between consecutive segments and it depends on the wingspan, such that

$$\xi = \frac{x_{N+1} - x_1}{N} . \quad (4.3)$$

After the partition, a discrete vortex is placed at the center of each segment in the wake, and its circulation is given by the integration of Eq. 4.1, from  $x_i$  to  $x_{i+1}$ , written as

$$\Gamma_i = \int_{x_i}^{x_{i+1}} -\frac{x'}{\sqrt{1-x'^2}} dx' . \quad (4.4)$$

In the present work, it is solved numerically by Gaussian quadrature either with 1 or 3 points.

Figure 4.6 shows that for  $n = 1$  the vortex particles are equally spaced, while for  $n > 1$ , they are concentrated towards the span extremities. Different values of  $n$ , being  $n = 1, 2$  and  $3$ , are tested in the simulations.

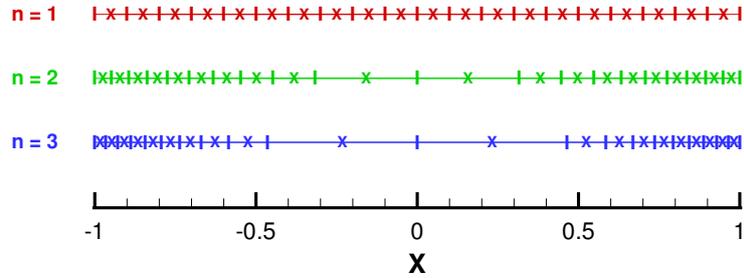


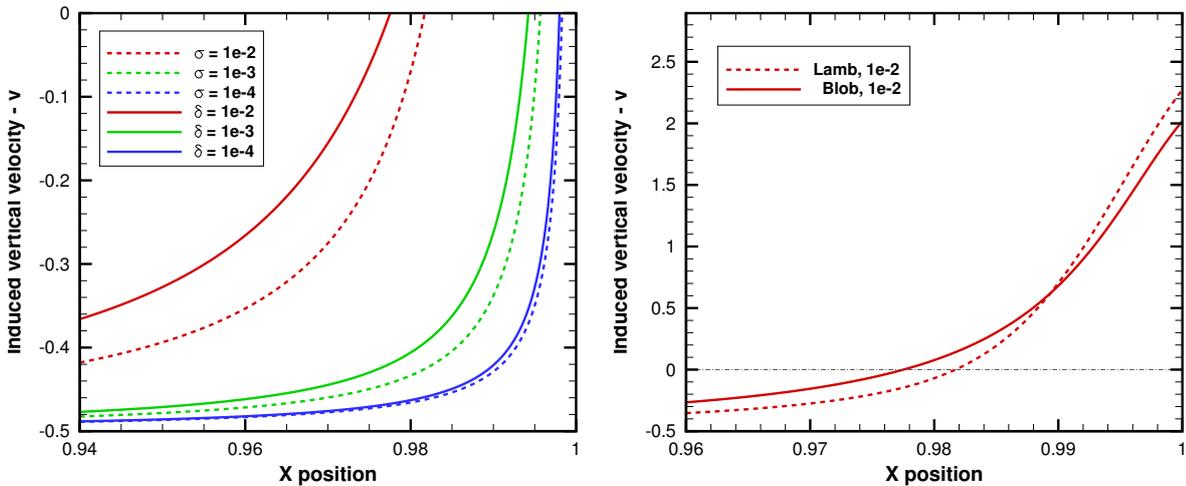
Figure 4.6: Different initial positions of the discrete vortex particles.

## 4.2 Results

Some initial definitions are required in order to solve the Trefftz plane problem, including the number of vortex particles in the wake discretization as well their initial position, since they can be equally spaced or concentrated towards the span tips (Eq. 4.2); quadrature scheme for the particles circulation based on Eq. 4.4. Also, the non-singular vortex model (Lamb or vortex Blob) must be defined, as well its viscous region. This section presents an investigation on all these parameters. If the Fast Multipole Method is applied to solve the velocity field, the maximum refinement level and the series truncation.

### 4.2.1 Vortex model

The induced vertical velocity from all vortex particles is calculated by a convolution from the Biot Savart law for different combinations of the variables above. Initially, the vortex models and their local diffusion are investigated for values of  $(\sigma, \delta)$  ranging from  $1 \times 10^{-2}$  to  $1 \times 10^{-4}$ , for 25k particles equally spaced. One should remind that the parameters  $\sigma$  and  $\delta$  refer to the sizes of the viscous core regions in the vortex Lamb and blob, respectively. Figure 4.7 shows the effects of potential vortex desingularization and one can see that the Lamb vortex model and smaller viscous cores induce larger velocities (in magnitude) compared to the vortex Blob model or more diffused viscous cores. In this figure, the dashed lines represent results obtained by the Lamb vortex model and continuous lines represent those obtained by the vortex blob model, for different values of viscous core  $\sigma$  and  $\delta$  indicated in the legend.



(a) Detailed view.

(b) Expanded view for  $\sigma = \delta = 0.01$ .

Figure 4.7: Induced vertical velocity for Lamb and vortex blob models.

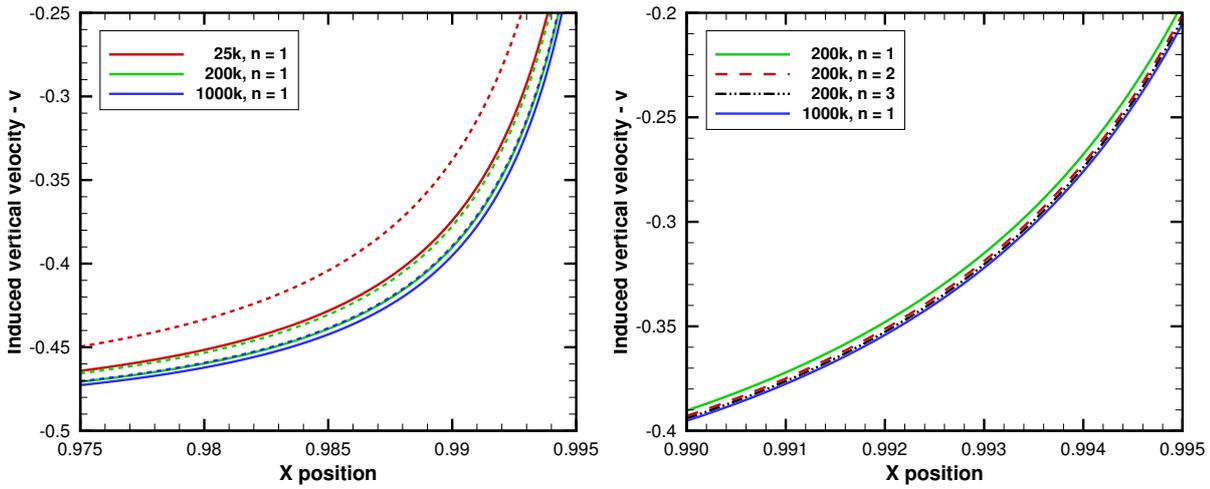
Unless specified, the Lamb vortex model will be employed in the forthcoming simulations since it has exponential convergence to the potential vortex model, as shown in Fig. 3.13. This characteristic enables the coupling of the DVM and FMM. Furthermore, the sharper velocity profile of the Lamb vortex (see Fig. 4.7) is more coherent with that expected by the inviscid potential vortex model. One should remind that the aim in this study is to perform simulations which better approximates the more extreme case of an inviscid flow.

## 4.2.2 Discretization

Results from the wake discretization are presented in Fig. 4.8(a) for  $N = 25\text{k}$ ,  $200\text{k}$  and  $1000\text{k}$  vortex particles equally spaced ( $n = 1$  in Eq. 4.2). Simulations are performed with a viscous core  $\sigma = 1 \times 10^{-3}$ . The accuracy in the circulation evaluated by numerical integration is also investigated. The dashed and solid lines represent, respectively, solutions obtained by 1 and 3 points in the Gaussian quadrature.

Concentrating the particles towards the tip increases the density of elements in the region where higher induced velocities are present without increasing the computational cost. Although there is a loss of precision in the wake center, the induced velocity profile in this region is smoother than that in the tips and the discretization does not require several elements. The solution obtained for an initial distribution of  $200\text{k}$  Lamb-vortex particles with  $\sigma = 1 \times 10^{-3}$  are presented in Fig. 4.8(b). The values of the exponent  $n$  in Eq. 4.2 are  $n = 1, 2$  and  $3$ . From this figure, one can observe that increasing the density of vortices towards the wing tip has the same effect as increasing the total number of elements with an equal distribution of vortex particles.

In Fig. 4.8(a), one can observe that more particles leads to a sharper induced velocity profile. The same effect can be noticed in Fig. 4.8(b) when the density of vortex particles is higher along the tips. This particular observation can be made from the comparisons between the velocity profile obtained for  $200\text{k}$  particles, for  $n = 3$ , and that computed for  $1000\text{k}$  particles. In conclusion, a higher density of particles in the wing tips (increasing the total number of elements or simply placing them towards the tip) and the calculation with 3 Gaussian quadrature points provide a better evaluation of the region near the singularity. Hence, the solution converges to a sharper velocity profile.



(a) Variation in the number of vortices.

(b) Variation in spatial distribution of vortices.

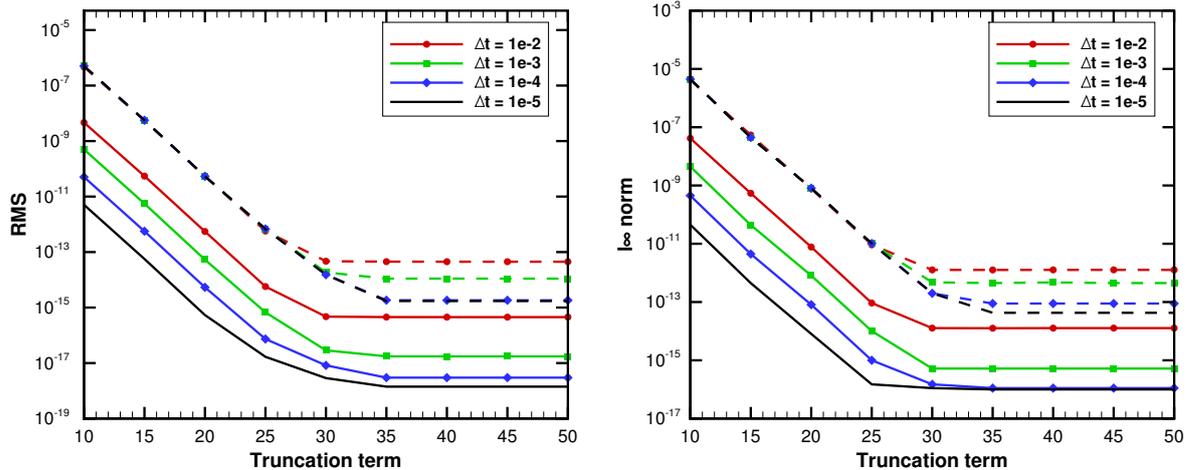
Figure 4.8: Effects of spatial distribution of vortex-particles in the wake discretization.

### 4.2.3 Fast Multipole Method parameters

The number of terms used in the FMM series expansion of the Green's function directly impacts the error of the numerical scheme and, therefore, the overall accuracy of the combination between the FMM and DVM. The solutions obtained by the application of the FMM coupled to the DVM are compared to those obtained solely by the DVM, which solves the Biot-Savart law. These solutions are compared through the root-mean-square, RMS, deviation as well the  $l_\infty$  norm for both velocity and vortex position. Here, comparisons are performed only for a single step of the simulation.

Figures 4.9 and 4.10 show the error behavior of the FMM as a function of the number of terms retained in the FMM series. Results are presented for the RMS and  $l_\infty$  norm of the error in terms of velocity (dashed lines) and vortex position (solid lines) after one convection step. Calculations are performed for 25k Lamb vortex with radius  $\sigma = 1 \times 10^{-3}$ .

In Fig. 4.9, one can see results of the error analysis computed for FMM refinement level 2. An assessment of the influence of the time step,  $\Delta t$ , on the error is presented for the fourth-order Runge-Kutta time marching scheme. It is possible to see that smaller time steps reduce errors in the FMM computation of both velocity and position. As expected, increasing the number of terms in the FMM series also reduces the error. One can also see that, for all values of time step, the error in position converges to the computer precision when 35 terms are used in the series. However, the velocity does not converge to the machine precision since several operations in the FMM suffer from truncation (factorials, Newton binomials and power series). It is worth to recall that the computer assures 15 digits of precision, while the 16<sup>th</sup> is rounded/truncated based on the 17<sup>th</sup>.



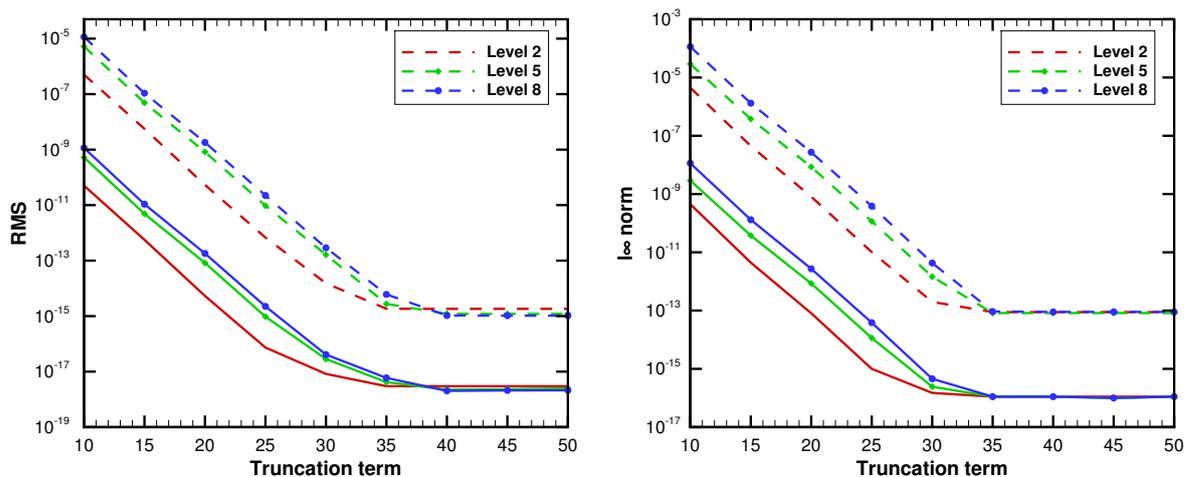
(a) RMS deviation for refinement level 2.

(b) Error  $l_\infty$  norm for refinement level 2.

Figure 4.9: Error analysis of FMM at refinement level 2 due to different time steps.

Figure 4.10 presents the error analysis for different levels of refinement of the FMM for a fixed time step of  $\Delta t = 1 \times 10^{-4}$ . One can see that higher levels of refinement produce larger errors when few terms are employed in the FMM series. However, double precision convergence is achieved for 40 truncation terms for all levels of refinement. Moreover, after convergence is achieved, the errors measured for the higher refinement levels are slightly smaller than those measured for the lower levels. Again, the  $l_\infty$  norm shows that the larger error in position is equal to the computer precision (16 digits) but the velocity presents to a larger error (13 digits).

One must not forget that dashed and solid lines indicate, respectively, the error for velocity and vortex position.



(a) RMS deviation for several refinement levels.

(b) Error  $l_\infty$  norm for several refinement levels.Figure 4.10: Error analysis of FMM at different levels for time step  $\Delta t = 1 \times 10^{-4}$ .

### 4.3 Vortex sheet roll-up

A study of long-time integrations for the vortex sheet roll-up is presented in this section, for both Biot-Savart direct summation and Fast Multipole Method. Also, the effects of several numerical parameters on the roll-up process are analyzed including: vortex model, time marching scheme, time step, size of viscous core, number of particles in wake discretization and computer precision.

#### 4.3.1 Biot-Savart computations

In Fig. 4.11, solutions using Biot-Savart direct calculation for the vortex sheet roll-up are compared for the vortex blob and Lamb vortex models for 25k particles placed towards the tip with a quadratic distribution,  $n = 2$ . The smoothing parameter is set as  $\delta = \sigma = 0.001$ . The velocity field is evaluated with the RK4 time marching scheme for 5000 time steps with  $\Delta t = 1 \times 10^{-5}$ . The circulation is computed by a Gaussian quadrature with one point. It is possible to see that the global structure of the roll-up is similar for both vortex models, although different initial velocity profiles are observed in Fig. 4.7.

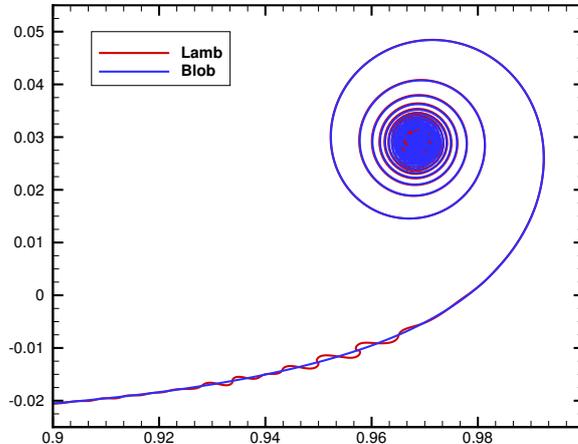


Figure 4.11: Comparison of vortex models in the sheet roll-up. Results are obtained with  $\delta = \sigma = 0.001$ , RK4,  $\Delta t = 1 \times 10^{-5}$ ,  $t^* = 0.05$ , 25k particles and  $n = 2$ .

Since the vortex blob model adds more diffusion to the solution, it delays the formation of spurious instabilities in the roll-up process. This same perturbation is present in computations by Krasny (1986a, 1987), and it was reduced by increased machine precision. On the other hand, the vortex blob smoothing parameter is purely numerical and lacks a physical approach. Moreover, it has a slow algebraic decay (compared to the fast exponential decay of the Lamb vortex model) which is not adequate for investigating the trends of compact vorticity field. Since this model has a poor compatibility with the FMM, it is discarded in the current work as explained in section 3.2.

In order to avoid this spurious perturbations, different time marching schemes were investigated, namely the RK1, RK2 and RK4, and the results are shown in Fig. 4.12. In the current simulations, the viscous core of the Lamb vortex is reduced to  $\sigma = 0.001$  since we want to assess the capability of the DVM to resolve sharper induced velocity profiles. Here, 25k particles are placed with a higher density towards the tip,  $n = 2$ , and calculations are performed with direct Biot-Savart evaluations using  $\Delta t = 1 \times 10^{-4}$ . As one can observe in Fig. 4.12, the RK1 scheme adds spurious diffusion to the solution (similar results are showed in section 2.2.6), which is responsible to delay the formation of the instabilities. However, the low accuracy of this time marching scheme compromises the solution along the roll-up region. On the other hand, the RK2 and RK4 schemes develop the instability but are able to resolve the roll-up region.

It is possible to notice that larger time steps also increase the numerical dissipation, damping hydrodynamic instabilities but compromising the wake roll-up, as shown in Fig. 4.13. Here, the RK4 scheme is employed and other parameters used in the simulations are similar to those from the analysis of the time marching scheme.

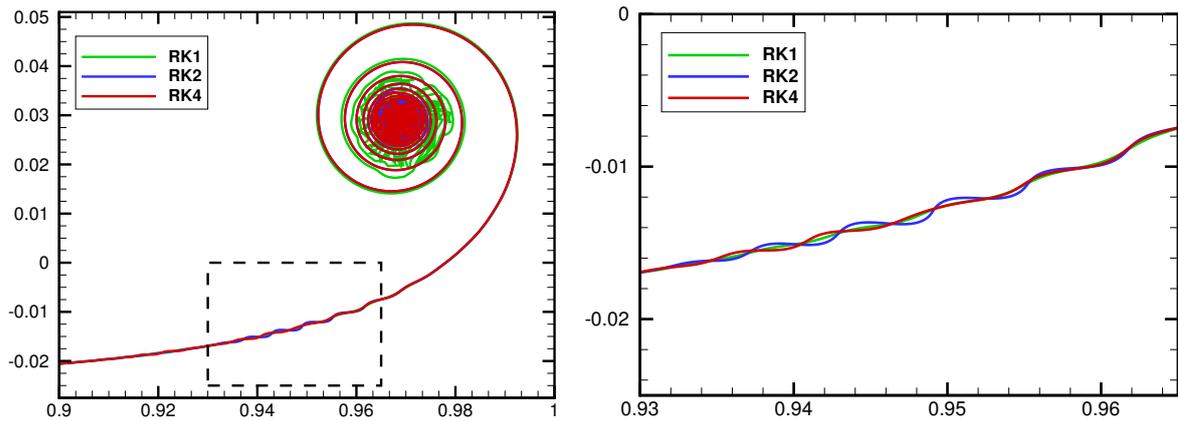


Figure 4.12: Comparison of time marching schemes for  $\sigma = 0.001$ ,  $\Delta t = 1 \times 10^{-4}$ ,  $t^* = 0.05$ ,  $n = 2$ , and 25k vortex particles.

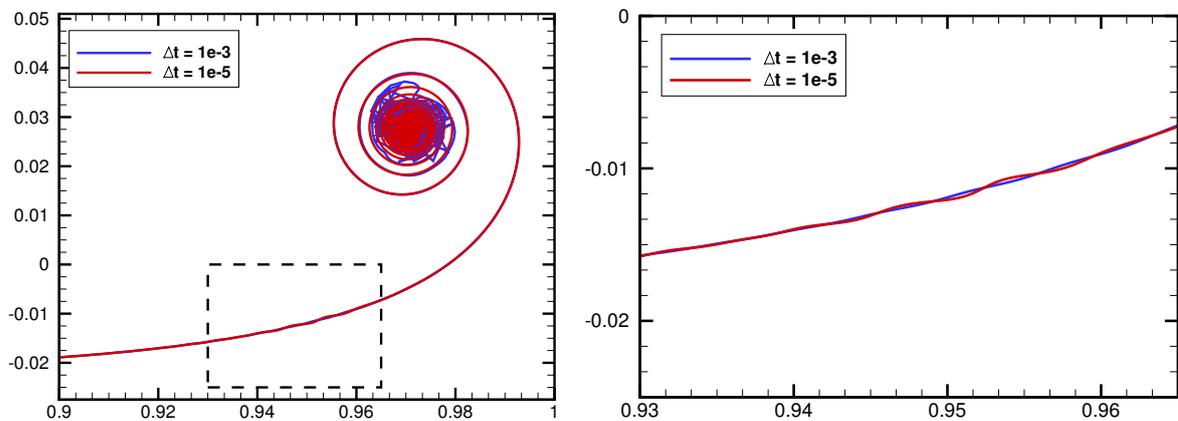


Figure 4.13: Time step effect for  $\sigma = 0.001$ , RK4,  $\Delta t = 1 \times 10^{-5}$ ,  $t^* = 0.045$ ,  $n = 2$  and 25k vortex particles.

### 4.3.2 Numerical accuracy of the FMM

As previously shown in section 4.2.3, for a single time-step, the FMM solution in terms of vortex particle position converges to that obtained by the direct calculation of the Biot-Savart law in double precision (8 bytes). In order to analyze the accumulated error of the FMM solution for several iterations, simulations are run with the fourth-order Runge-Kutta scheme for 10k time steps with  $\Delta t = 1 \times 10^{-5}$ . The Lamb vortex model is employed with  $\sigma = 0.001$  for a wake discretized by 25k particles, initially placed in a quadratic distribution along the wake ( $n = 2$ ). These parameters are the same as those simulated before, with the only difference on the method used to solve.

Although initially the  $l_\infty$  norm is basically due to the machine precision for a few time steps, the error accumulates. The results are shown in Fig. 4.14 for double precision (8 bytes). After several steps, the deviation of both methods becomes significant.

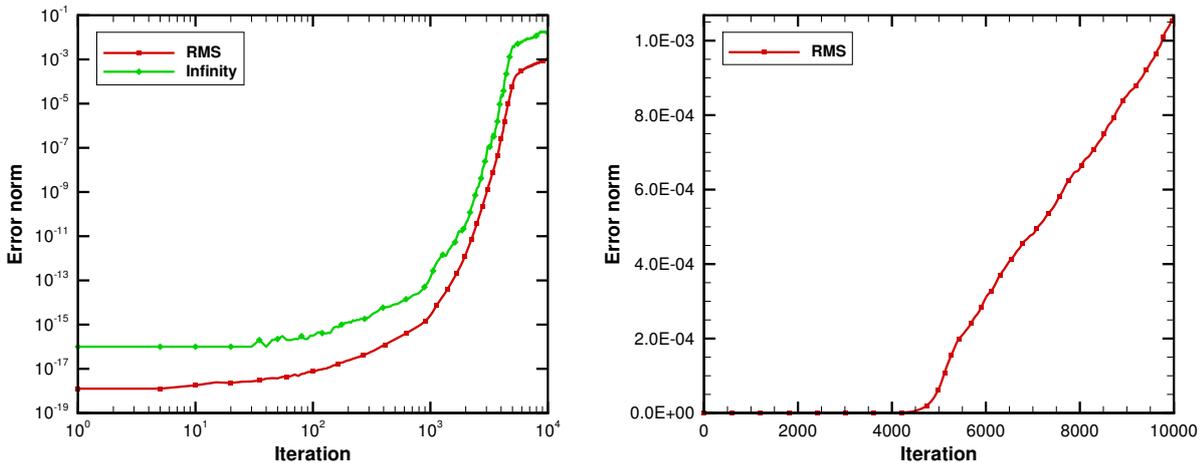
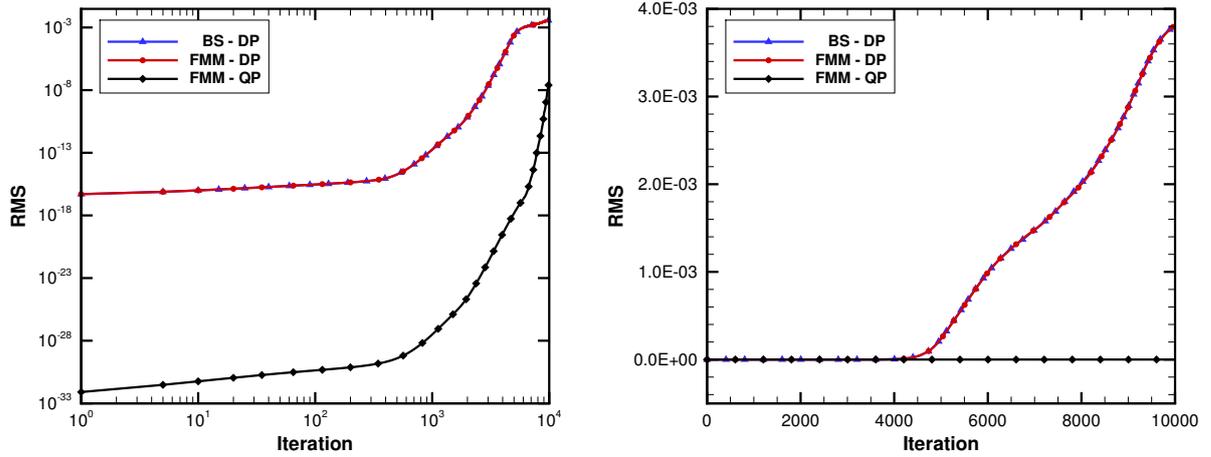


Figure 4.14: Error from FMM to Biot-Savart direct summation.

These measurements were performed also in quadruple precision (16 bytes), for both FMM and Biot-Savart, and the accumulated error is shown in Fig. 4.15. Although the fast method has an error to Biot-Savart in double precision, Fig. 4.14, when both are compared to the Biot-Savart in quadruple precision, they have the same behavior. This indicates that the error measurement in double precision is purely due to machine truncation. Therefore, with the FMM it is possible to obtain an *artificial exact solution* compared to the direct calculation of the Biot-Savart law by the DVM.

Furthermore, if the FMM is implemented using QP, one can achieve considerably lower errors compared to Biot-Savart in DP. Here, the errors are computed using the RMS norm based on the QP solution of the direct calculation of the Biot-Savart law through the DVM. The effects of the long-time error on the physical solution of the Trefftz plane are discussed in the next sections which describe the vortex sheet roll-up characteristics.



(a) RMS deviation in terms of particle position (log scale).

(b) RMS deviation in terms of particle position (linear scale).

Figure 4.15: Long-term truncation error for FMM and Biot-Savart law.

### 4.3.3 Fast Multipole Method computations

From the present results, the spurious instabilities are inherent from the DVM solution with reduced viscosity. With the FMM-DVM coupling, it is possible to assess the effects of increasing the number of particles in the cloud to suppress this spurious instability.

Results are shown in Fig. 4.16 for calculations performed with the RK4,  $\sigma = 0.001$ ,  $\Delta t = 1 \times 10^{-5}$ ,  $n = 2$  for  $t^* = 0.045$ . Here, the FMM is employed with the DVM and finer discretization leads to better resolutions with delay in the formation of the instabilities and as well smoother roll-up regions.

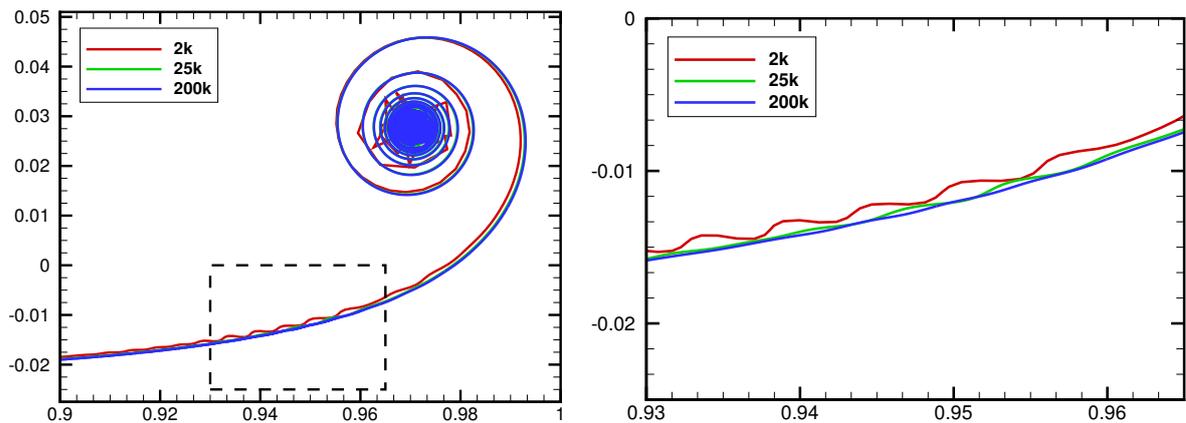


Figure 4.16: Discretization effects for different number of particles with  $\sigma = 0.001$ , RK4,  $\Delta t = 1 \times 10^{-5}$ ,  $t^* = 0.045$  and  $n = 2$ .

Since the FMM allows a higher resolution using larger number of vortex particles in the wake discretization, it is possible to reduce the core  $\sigma$  and still maintain a continuous shear layer, *i.e.*, the distance between two vortices should still be lower than the viscous core. In this sense, Fig. 4.17 presents a study of the dynamics of the roll-up. Solutions are obtained by the FMM-DVM for 200k vortex particles concentrated towards the tip with  $n = 3$  convected by the RK4 for 75 time steps with  $\Delta t = 4 \times 10^{-7}$ .

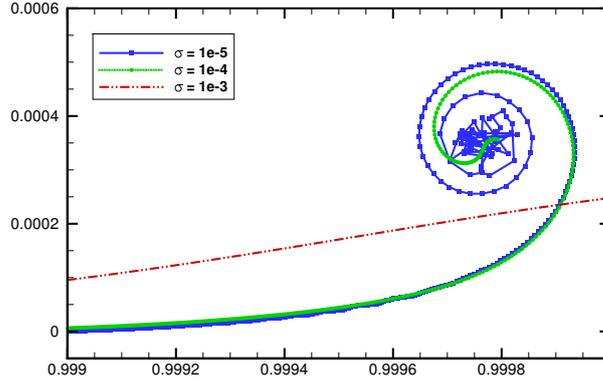


Figure 4.17: Roll-up process for different viscous cores,  $\sigma$ . Solutions obtained with 200k vortex particles, RK4,  $\Delta t = 4 \times 10^{-7}$ ,  $t^* = 3 \times 10^{-5}$  and  $n = 3$ .

Results shown in Fig. 4.17 demonstrate that smaller viscous cores lead to a faster roll-up, in agreement with Fig. 4.7(a). For example, when  $\sigma = 1 \times 10^{-3}$ , the vortex sheet does not begin the roll-up process and, for  $\sigma = 1 \times 10^{-5}$ , it shows an advanced stage of the roll-up with a disorganized core. Furthermore, there is also a spurious perturbation of the sheet in this last case.

The effects of time step are investigated for a small viscous core,  $\sigma = 1 \times 10^{-5}$ , and results are shown in Fig. 4.18 for the RK4 and 200k particles at  $t^* = 3 \times 10^{-5}$ . As expected, smaller time steps lead to better roll-up structures. However, even for  $\Delta t = 8 \times 10^{-8}$ , there is still a chaotic pattern in the core. Since the viscous core is very small, the dissipation of the larger time-step is not enough to show effects on suppressing the spurious instability. This test case and those below are performed using  $n = 3$  to increase the particle density towards the tips of the wake. Gaussian quadrature with 3 points is used to compute the vortices circulation.

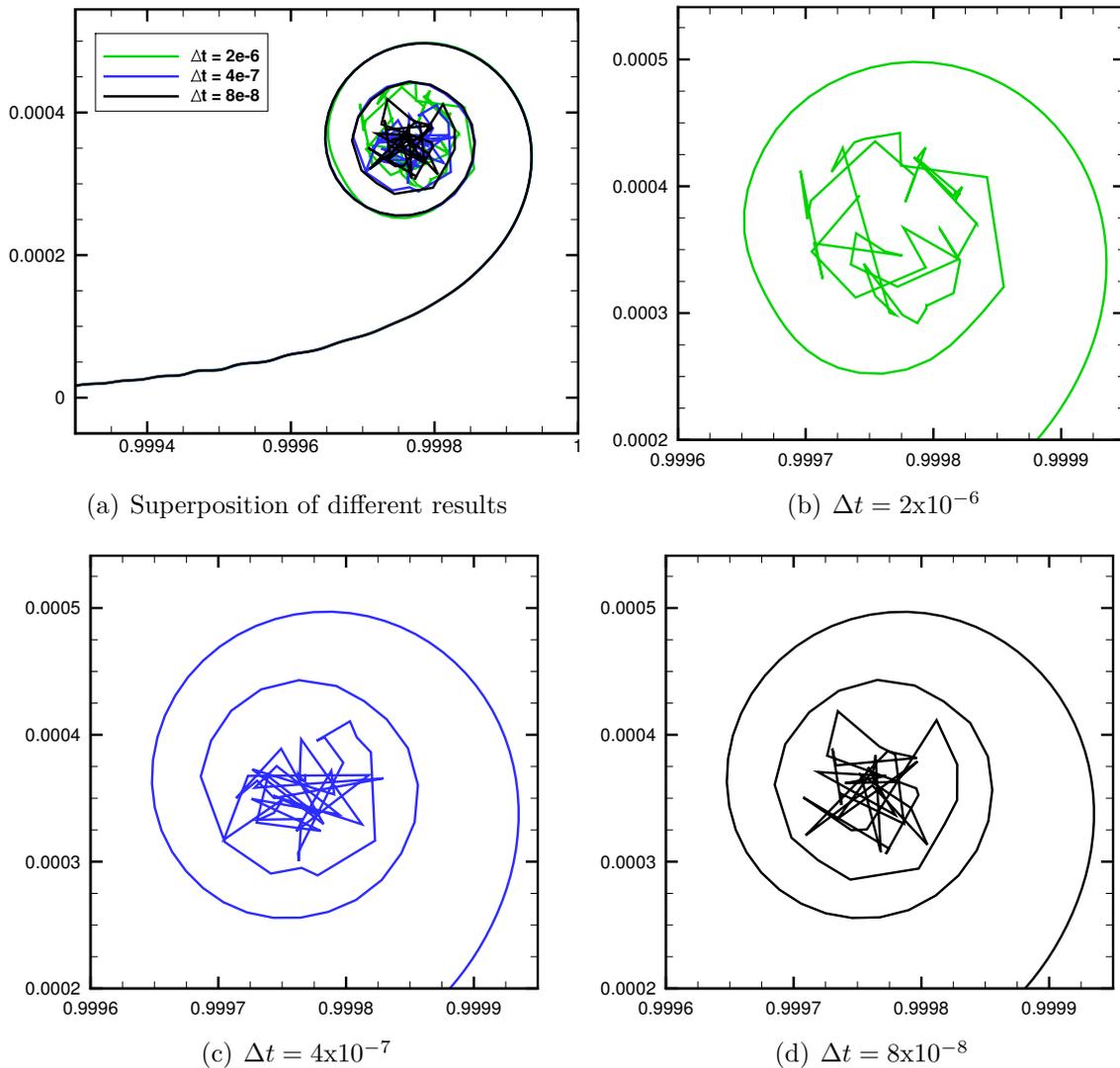


Figure 4.18: Time step influence on roll-up process for 200k vortex particles,  $\sigma = 1 \times 10^{-5}$ , RK4,  $t^* = 3 \times 10^{-5}$ ,  $n = 3$  and Gaussian quadrature with 3 points.

With the computational cost reduction from the FMM, it is possible to refine the discretization with more vortex particles. Figure 4.19 shows the effects of increasing the number of particles in the roll-up solution. Computations with up to 1 million Lamb-Oseen vortex particles with  $\sigma = 1 \times 10^{-5}$  and  $\Delta t = 8 \times 10^{-8}$  for 375 time steps. As it is clear from this figure, more particles are required to suppress the instabilities. Also, when smaller cores are used, the roll-up in the vortex core is easily destabilized and the introduction of more particles improve the solutions.

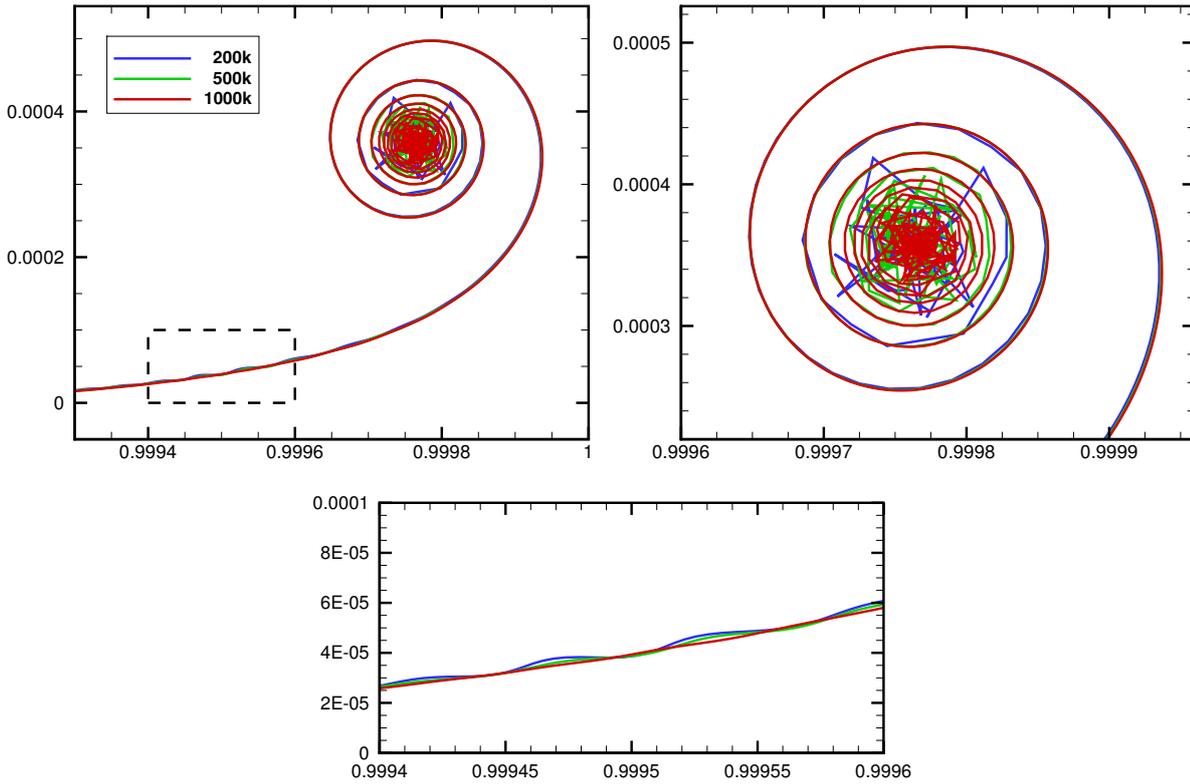


Figure 4.19: Number of particles influence for  $\sigma = 1 \times 10^{-5}$ , RK4,  $\Delta t = 8 \times 10^{-8}$ ,  $t^* = 3 \times 10^{-5}$ ,  $n = 3$  and Gaussian quadrature with 3 points.

Figure 4.20 shows that the instabilities in the vortex layer are purely numerical since they can be delayed by increasing the precision in calculations. The numerical error is reduced using quadruple precision and, as one can see in this figure, the solution is free of instabilities when higher computer precision is employed. A convergence test performed for quadruple precision indicated that  $p = 75$  terms in the power series of the FMM provide quadruple machine precision. These simulations are run for 200k particles and 375 steps using RK4 with  $\Delta t = 8 \times 10^{-8}$ .

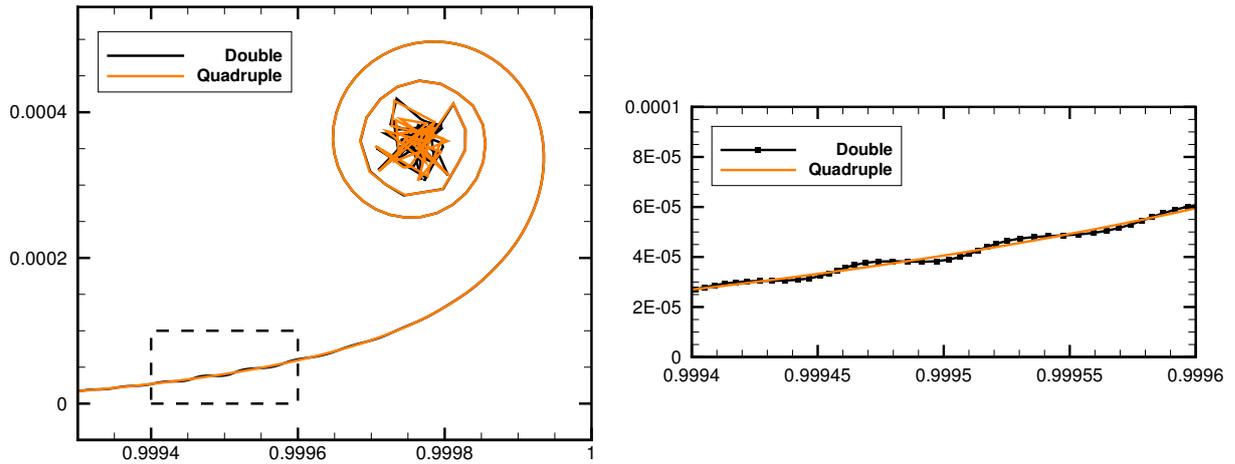


Figure 4.20: Influence of machine precision on suppression of instabilities for 200k vortex,  $\sigma = 1 \times 10^{-5}$ , RK4,  $\Delta t = 8 \times 10^{-8}$ ,  $t^* = 3 \times 10^{-5}$ ,  $n = 3$  and quadrature with 3 points.

Lastly, the effects of diffusion are assessed through the implementation of the core spreading method, CSM, to solve the diffusion term in the vorticity equation. When this method is implemented, all particles are initially singular, with  $\sigma = 0$ , and, at every iteration, they diffuse according to the Reynolds number and time step (see section 2.3.1 for details). Although the solution does not converge for long time integration, as shown by Greengard (1985), it is still valid for short time as explained by Kida and Nakajima (1998). For this test,  $\nu = 1 \times 10^{-5}$  and  $\Delta t = 8 \times 10^{-8}$ . To convect the 1000k vortices, simulations are run with the RK4 for 450 iterations. Therefore, all vortex particles have  $\sigma \approx 3.79 \times 10^{-5}$  at the end of the calculations, based on Eq. 2.81.

In Fig. 4.21, it is possible to see the absence of instabilities when the viscous term of the equation is solved. Hence, the lack of viscosity, a natural mechanism for dissipation, leads to amplification of numerical errors in the inviscid case.

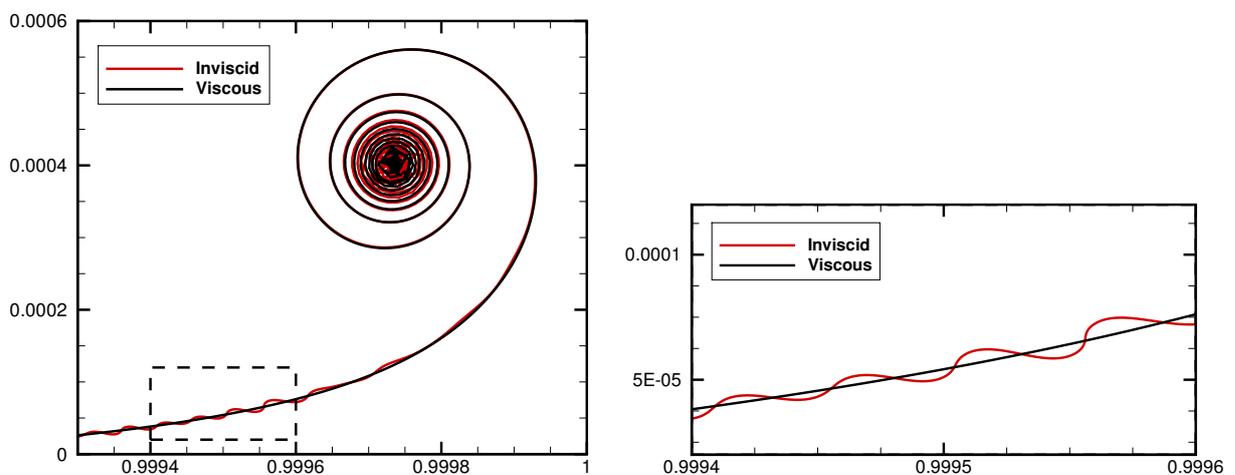


Figure 4.21: Viscosity influence for 1000k vortex,  $\sigma = 1 \times 10^{-5}$ , RK4,  $\Delta t = 8 \times 10^{-8}$ ,  $t^* = 3.6 \times 10^{-5}$ ,  $n = 3$  and Gaussian quadrature with 3 points.

### 4.3.4 Computational time

The FMM provides two main benefits when coupled to the DVM. It reduces the computational cost and, in consequence, it allows simulations with larger numbers of particles and, therefore, higher resolutions in the discretization of the physical problems of interest.

Here, we perform comparisons to quantify the savings in computational cost of the coupled FMM-DVM compared to the direct solution of the Biot-Savart law, which has a cost purely proportional to  $O(N^2)$ . Comparisons are performed for a single evaluation of the velocity field induced by interactions among all particles. An assessment of the computational cost is presented for different refinement levels and number of vortex particles in the cloud. All computations are performed using  $p = 40$  terms in the FMM series expansion, which guarantees computer accuracy in double precision. The present simulations are performed in serial by a 2.5 GHz Intel® Xeon™ E5-2670v2 CPU.

Figure 4.22 shows a comparison in terms of computational wall time for the direct evaluation of the Biot-Savart and the FMM. It is possible to see that, for the number of particles analyzed, the latter provides faster solutions than the former, except for higher levels of refinement with a number of particles below 4000. Simulations are performed using up to  $5 \times 10^5$  particles and, for this limit case, time savings of 285 times are achieved by the fast method. Also, it is interesting to notice that, for each level of refinement of the FMM, the computational cost is constant until a certain number of particles is reached. The fixed cost is dominant so higher plateaus are observed for finer maximum refinement levels. After the plateau, the cost behavior is also proportional to  $O(N^2)$  but, higher refinement levels have shifted curves which provide lower computational costs. This leads to intersections of the curves, so, after a number of elements large enough, higher refinement levels are required to avoid excessive computational cost.

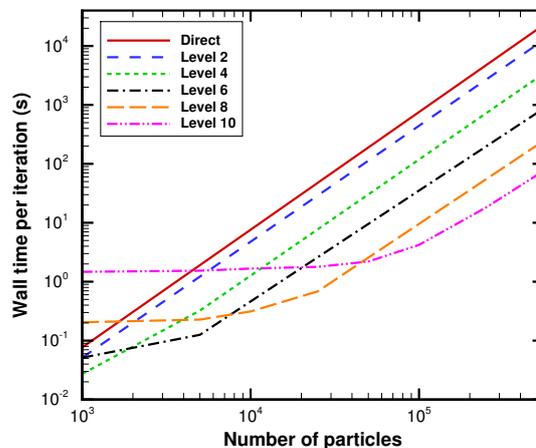
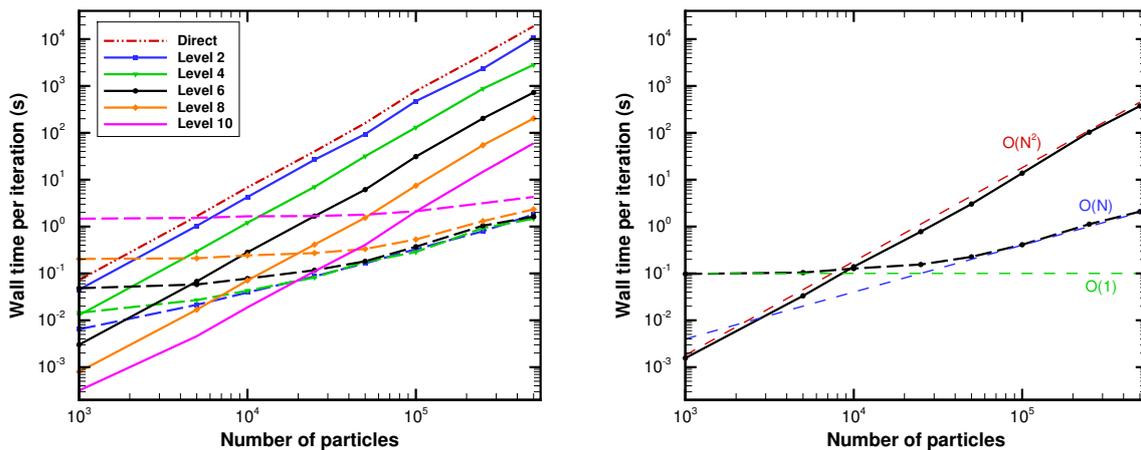


Figure 4.22: Computational time to evaluate the velocity field for Biot-Savart and FMM.

Furthermore, Fig. 4.22 shows that, as the number of particles increases, higher levels of refinement are required to avoid excessive computational cost per iteration. This confirms the necessity to apply further refinements of the FMM domain in problems with vorticity generation, as the one presented in section 3.2. Hence, given a number  $N$  of elements, there is an excellent refinement level that keeps the computational time at a minimum.

In order to understand the format of these curves, Fig. 4.23(a) shows the comparison between near-field and far-field costs of the FMM algorithm, respectively, solid and dashed lines. The total computational cost of the FMM is given by the sum of these two contributions and it should be of the form  $O(\mathcal{A}_2 N^2 + \mathcal{A}_1 N + \mathcal{A}_0)$ , as explained in Eq. 3.3. The computational cost of the near-field calculations is still  $O(\mathcal{A}_2 N^2)$  since it represents the direct evaluations of the Biot-Savart law performed for vortex particles contained inside neighbor boxes of the FMM. For increased refinement levels, the near-field interaction cost is smaller since there are less direct evaluations at finer boxes, *i.e.*, the coefficient  $\mathcal{A}_2$  is reduced. This way, the cost of the solely direct evaluation is reduced. On the other hand, higher levels of refinement increase the coefficients  $\mathcal{A}_1$  and  $\mathcal{A}_0$  since more FMM operations are required, which leads to higher far-field costs.

Figure 4.23(b) shows the cost for near and far-field operations of the FMM for refinement level 7. Aside from the constants  $\mathcal{A}$ , one can see that the near-field behavior is proportional to  $O(N^2)$  while that for the far-field is a combination of  $O(1)$  and  $O(N)$ . For a fixed refinement level, if  $N$  is very large, the dominant cost is  $O(N^2)$ , while for small number of elements, the fixed cost  $O(1)$  is dominant. Also, the linear cost  $O(N)$  (P2M and L2P steps) is small compared to the other costs, indicating that near-field cost (P2P) and box-box operations (M2L) are dominant.



(a) Wall time for several levels.

(b) Wall time for refinement level 7.

Figure 4.23: Computational time to evaluate far-field and near-field operations.

In Fig. 4.24, one can observe a comparison in terms of computational cost for the FMM implemented using quadruple precision (QP) and for the direct evaluation of the Biot-Savart law using DP and QP. The FMM solution provides a lower computational error than the DP direct evaluation as shown in Fig. 4.15. Moreover, for a large number of particles, it is faster than the DP direct evaluation as one can see in Fig. 4.24.

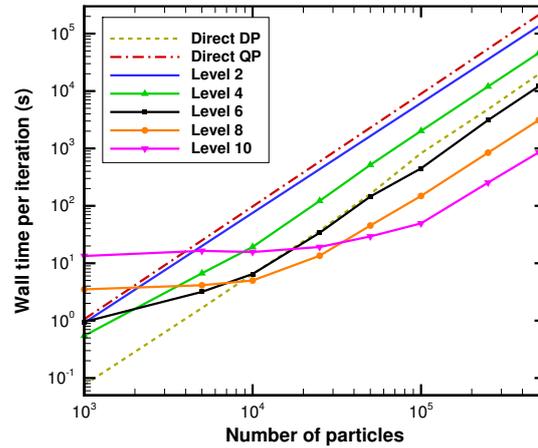


Figure 4.24: Influence of quadruple precision for both Biot-Savart and FMM computational time per step.

## 5 PERIODIC SHEAR LAYER

When two fluids share an interface and are flowing with different velocities, a line of vorticity is generated forming a shear layer. Any perturbation creates waves with an associated length  $\lambda$  that amplifies in time. This phenomenon is known as the Kelvin-Helmholtz instability and it is shown in Fig. 5.1. This kind of instability occur in several problems of engineering and physics, for example, in ocean mixing, cloud formation, multi-phase flow, combustion and aircraft jets. A review on the topic, with further consultation material, is presented by Matsuoka (2014).

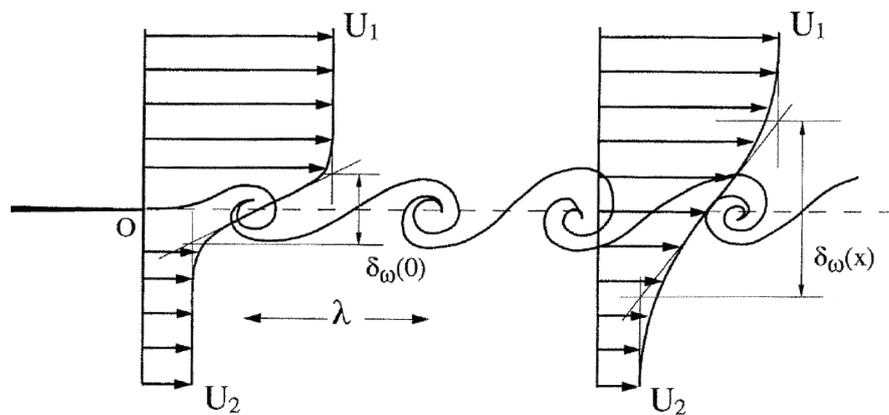


Figure 5.1: Kelvin-Helmholtz instability present on mixing-layers.

### 5.1 Numerical formulation

The discrete vortex method can be employed to study the evolution of shear layers including the formation of the Kelvin-Helmholtz instability. With the DVM, a periodical shear layer can be discretized using  $N$  vortex-particles and its temporal evolution can be simulated depending on the imposed initial conditions. Since the DVM cost is proportional to  $O(N^2)$ , it is convenient to use the fast multipole method to reduce the overall simulation time. However, further modifications need to be performed to the free-domain FMM implementation in order to apply the method for the solution of periodical problems. The application of the FMM for the solution of periodical problems is reported by several authors and, for more details, one should see Kabadshow (2010); Kudin and Scuseria (1998); Lambert *et. al* (1996); Kurzak and Pettitt (2006); Rodin and Overfelt (2004); Yokota and Obi (2010). This chapter presents implementation details of the periodical FMM including its application to study the time evolution of a periodical shear layer.

The discretization of a periodical shear layer using potential vortices in the DVM may lead to unrealistic velocities when particles come close to each other. Hence, in this chapter, models for non-singular vortex-particles are studied to obtain smooth velocity fields, coherent with the physics of the problem.

### 5.1.1 Initial conditions and discretization

The vorticity  $\gamma$  of a continuous shear layer with length  $\lambda$  is given by

$$\gamma = \oint \mathbf{u} \cdot d\mathbf{s} , \quad (5.1)$$

where, from Fig. 5.1,

$$\gamma = U_1\lambda - U_2\lambda = \Delta U\lambda . \quad (5.2)$$

Discretizing the shear layer using  $N$  vortex elements with equal length  $\lambda/N$  leads to

$$\Gamma_m = \oint_{\lambda/N} \mathbf{u} \cdot d\mathbf{s} = \frac{\Delta U\lambda}{N} . \quad (5.3)$$

Setting  $\Delta U = 1$  and  $\lambda = 1$ , the circulation of each vortex is given by

$$\Gamma_m = \frac{1}{N} . \quad (5.4)$$

If the periodic vorticity layer is initially straight, without perturbations, by symmetry of the problem, its self-induced velocity is null and there is no motion. However, computer truncation adds a small random perturbation to the shear layer and spurious waves arise. In order to create a controlled solution, we add a sinusoidal perturbation with length  $\lambda$  and amplitude  $A$ ,  $A \ll \lambda$ , as shown in Fig. 5.2.

$$y_m = A \sin\left(\frac{2\pi}{\lambda}x_m\right) , \quad (5.5)$$

where

$$-0.5 < x_m < 0.5 . \quad (5.6)$$

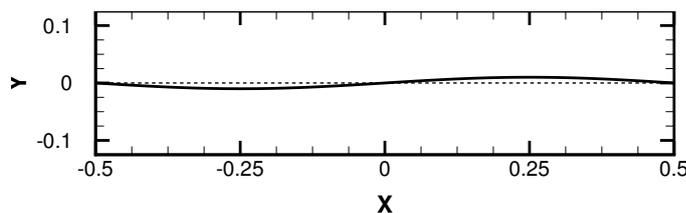


Figure 5.2: Sinusoidal initial perturbation, with amplitude  $A = 0.01$ , of a shear layer.

## 5.1.2 Induced velocity by a vortex array

The induced velocity at a point  $z_i$  by a vortex-particle at  $z_j$ , with  $\Gamma_j > 0$  in clockwise sense, is given by the Biot-Savart law shown in Eq. 2.44 of section 2.2.3 as:

$$w_i = i \frac{\Gamma_j}{2\pi z_{ij}} ,$$

where

$$w_i = u_i - i v_i ,$$

and

$$z_{ij} = z_i - z_j = (x_i - x_j) + i(y_i - y_j) .$$

If the vortex  $j$  is replicated  $n$  times, from  $-\infty$  to  $\infty$ , with spatial periodicity  $\lambda$ , it is possible to calculate the influence of this array at  $z_i$  according to

$$w_i = \sum_{n=-\infty}^{\infty} i \frac{\Gamma_j}{2\pi} \left( \frac{1}{z_{ij} + n\lambda} \right) . \quad (5.7)$$

For  $\lambda = 1$ , this equation can be simplified to

$$w_i = i \frac{\Gamma_j}{2\pi} \left[ \frac{1}{z_{ij}} + \sum_{n=1}^{\infty} \left( \frac{1}{z_{ij} - n} + \frac{1}{z_{ij} + n} \right) \right] , \quad (5.8)$$

where the summation inside parentheses has an exact solution given by the cotangent function (Abramowitz and Stegun, 1964)

$$\pi \cot(\pi z_{ij}) = \frac{1}{z_{ij}} + \sum_{n=1}^{\infty} \left( \frac{1}{z_{ij} - n} + \frac{1}{z_{ij} + n} \right) . \quad (5.9)$$

This leads to an exact solution for the induced complex velocity by an infinite array of periodical vortex particles  $j$  at a point  $z_i$ . Therefore, it is possible to rewrite Eq. 5.8 as

$$w_i = i \frac{\Gamma_j}{2\pi} \left[ \pi \cot(\pi z_{ij}) \right] . \quad (5.10)$$

The cotangent solution can be manipulated using trigonometrical relations for an arbitrary complex argument  $z$  given by

$$\cot(z) = i \coth(iz) , \quad (5.11)$$

$$\sinh(z) = -i \sin(iz) , \quad (5.12)$$

$$\cosh(z) = + \cos(iz) , \quad (5.13)$$

such that

$$\coth z = \frac{\sinh(2x) - i \sin(2y)}{\cosh(2x) - \cos(2y)} , \quad (5.14)$$

and, using the following even-odd properties of functions

$$\sin(-z) = -\sin(z) , \quad (5.15)$$

$$\cos(-z) = +\cos(z) , \quad (5.16)$$

it is possible to evaluate the  $u$  and  $v$  components in their final decomposed form as, respectively,

$$u_i = \frac{\Gamma_j}{2\pi} \left( \frac{+\pi \sinh(2\pi y_{ij})}{\cosh(2\pi y_{ij}) + \cos(2\pi x_{ij})} \right) , \quad (5.17)$$

$$v_i = \frac{\Gamma_j}{2\pi} \left( \frac{-\pi \sin(2\pi x_{ij})}{\cosh(2\pi y_{ij}) + \cos(2\pi x_{ij})} \right) , \quad (5.18)$$

where

$$x_{ij} = x_i - x_j \quad (5.19)$$

and

$$y_{ij} = y_i - y_j . \quad (5.20)$$

### 5.1.3 Periodic FMM

For a potential evaluated at  $z_o$  due to a source placed at  $z_s$ , one can write the approximation

$$\phi(z_o) = \mathcal{M}_0 \log(z) + \sum_{k=1}^{\infty} \frac{\mathcal{M}_k}{z^k} , \quad (5.21)$$

where the multipoles  $\mathcal{M}_0$  and  $\mathcal{M}_k$  are defined according to Eqs. 3.17 and 3.18, respectively. The distance on the complex plane is given by

$$z = z_o - z_s . \quad (5.22)$$

In their paper, Greengard and Rokhlin (1987) show different boundary conditions for the FMM that can be applied for multi dimensional problems. If one uses periodical boundary conditions, where the sources are replicated with spatial periodicity  $\lambda$ , the distance of the  $n$ -th source to the observer is defined as

$$z_n = z_o - (z_s + n\lambda) , \quad n \in \mathbb{Z} , \quad (5.23)$$

where it is necessary to evaluate the potential  $\phi(z_o)$  due to sources at  $|z_n| \rightarrow \infty$ . This is not possible for finite charges,  $\mathcal{M}_0 \neq 0$ , since  $\log(z)$  is unbounded for  $|z| \rightarrow \infty$ . Moreover, the derivative of the potential, which is the term computed in the DVM, can be written as

$$w(z_o) = \frac{\partial \phi(z_o)}{\partial z_o} = \frac{\mathcal{M}_0}{z} + \sum_{k=1}^{\infty} -k \frac{\mathcal{M}_k}{z^{k+1}} . \quad (5.24)$$

This term is bounded everywhere, even for  $|z| \rightarrow \infty$  and, therefore, derivatives for periodic problems with finite charges can be solved.

The implementation of periodic boundary conditions in the FMM is straightforward and it will be explained in one dimension, but can be generalized for multiple dimensions. In a one dimensional problem, the velocity induced by an infinite array of vortex particles with spatial periodicity  $\lambda = 1$  at the observer  $z_o$ , shown in Eq. 5.7, is

$$w(z_o) = \sum_{n=-\infty}^{\infty} i \frac{\Gamma_s}{2\pi} \left( \frac{1}{z+n} \right) ,$$

where  $z$  is the same distance of the free-domain FMM, given by Eq. 5.22.

The convolution kernel can be rewritten as

$$\sum_{n=-\infty}^{-2} \left( \frac{1}{z+n} \right) + \left( \frac{1}{z-1} \right) + \left( \frac{1}{z} \right) + \left( \frac{1}{z+1} \right) + \sum_{n=2}^{\infty} \left( \frac{1}{z+n} \right) . \quad (5.25)$$

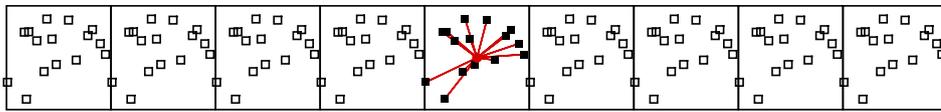
Each term  $n$  in the summation of Eq. 5.25 represents a periodic box, at level  $\ell = 0$ , in the infinite array. These periodical boxes  $n$  contain the periodical particles from the initial box denominated *central domain*,  $n = 0$ . One can associate Eq. 5.25 to the following FMM structure:

$$\begin{aligned} \sum_{n=-\infty}^{-2} \left( \frac{1}{z+n} \right) & : \text{ well-separated boxes on the left side of the central domain,} \\ \left( \frac{1}{z-1} \right) & : \text{ left neighbor at level 0 of the central domain,} \\ \left( \frac{1}{z} \right) & : \text{ central domain at level 0,} \\ \left( \frac{1}{z+1} \right) & : \text{ right neighbor at level 0 of the central domain,} \\ \sum_{n=2}^{\infty} \left( \frac{1}{z+n} \right) & : \text{ well-separated boxes on the right side of the central domain.} \end{aligned}$$

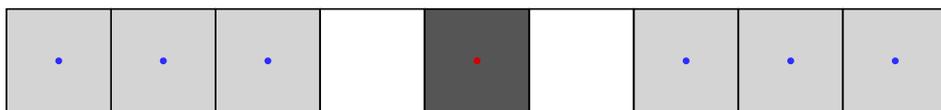
Due to periodicity, all clusters of particles at level  $\ell = 0$  are identical, which implies that the multipole expansions for all periodic boxes, relative to the respective centroids, are the same. Hence, instead of creating several domains, it is possible to simply replicate the level 0 box central domain in both the left and right directions. Therefore, one

can avoid any extra cost for the multipole-to-multipole (M2M) upward pass from higher refinement levels,  $\ell > 0$ , in the replicated boxes. This procedure is shown in Fig. 5.3(a), where the particles indicated by non-filled squares are not accounted for at first, only those indicated by black dots in the central domain.

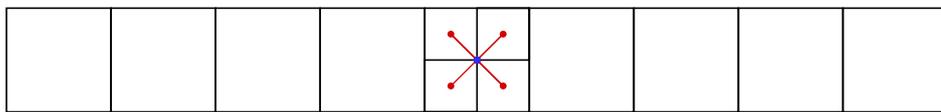
The far-field interactions at level 0 are possible because both left and right periodic images, indicated by  $|n| \geq 2$  to  $\infty$ , are *well-separated*. The number of periodic domains has to be chosen according to the desired precision in the calculations. They interact once with the central box using multipole-to-local (M2L) operations at level 0. This is illustrated in Fig. 5.3(b), where the light gray boxes interact with the central dark gray box. After this operation, the translations to all four children at level 1 are performed using the local-to-local (L2L) step shown in Fig. 5.3(c). However, the particles inside both left and right neighbors, as well as those inside the central domain, only interact at levels  $\ell > 0$  since their respective boxes are not well-separated. These interactions are computed in step 4 for free-domain FMM (see section 3.1.2 for details). The operations are shown in Fig. 5.3(d), where for light and dark gray boxes indicate source and observer boxes respectively. To do so, modifications in the interaction and neighbor lists are required as they have to account for the smaller boxes at levels  $\ell > 0$  in both left and right periodic neighbors ( $|n| = 1$ ).



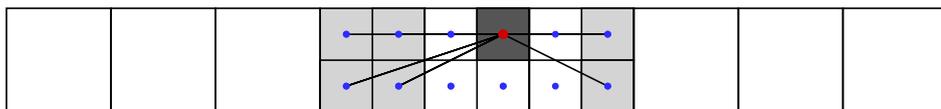
(a) M2M is performed up to level 0 only for the central box, ignoring all the other image boxes.



(b) M2L operations are performed at level 0 for far-field replicated images.



(c) L2L from level 0 to 1 is performed only for the four children of the central box.



(d) M2L step for one of the 4 children of the central box, at level 1, is computed using near-field replicated images.

Figure 5.3: Schematic of operations performed in the periodic FMM.

The divide-and-conquer strategy can be further explored to reduce the interaction costs among clusters at level 0, and, to do so, an additional upward pass using M2M computation can be employed. The basis of this optimized scheme is that periodic boxes  $|n| \geq 2$  can be further grouped to create large clusters. One must not forget that, from the separation criterion, it is only mandatory that the clusters are separated by at least the size of the source-boxes in order to perform M2L operations. These new clusters are larger than that of the central domain. In the free-domain FMM context, it is possible to say that they are equivalent to negative levels  $\ell$  of refinement in Eq. 3.7, copied below

$$S_\ell = \frac{S_0}{2^\ell} .$$

However, a change in nomenclature is used for easiness of implementation: a *periodic level*  $i$  is defined instead of working with negative refinement levels. This modified methodology works with positive values, *i.e.*, levels  $i \geq 0$ . The maximum refinement level  $I$  indicates the number of periodic domains and is chosen based on the desired accuracy.

In this work, a cluster  $\mathcal{P}_i$ , at the periodic level  $i$ , contains a number of baseline-boxes (level 0 boxes) equal to the  $i$ -th power of 2, for  $i \geq 0$ , as shown by Eq. 5.26. These cluster combinations are performed using M2M operations from level  $i = 0$ , which coincides with a single box at level  $\ell = 0$  of the free-domain FMM, up to the maximum periodic level  $I$ . After the clustering operations, the new large cluster  $\mathcal{P}_i$  may perform M2L operations to evaluate its influence in the central domain. One must note that higher periodic levels  $i$  are associated to the more distant, larger clusters of boxes.

$$\mathcal{P}_i = 2^i \quad \text{or} \quad i = \log_2 \mathcal{P}_i . \quad (5.26)$$

For a single direction along the  $x$ -axis, the total number of well-separated baseline-boxes which interacts with the central domain, up to the maximum periodic level  $I$ , is

$$\sum_{i=0}^I \mathcal{P}_i . \quad (5.27)$$

Furthermore, if one includes the domain  $n = 1$ , which does not interact directly with the central domain via FMM at level 0, the total number of baseline-boxes along one direction of the  $x$ -axis is

$$1 + \sum_{i=0}^I \mathcal{P}_i , \quad (5.28)$$

and one can show by induction that the summation of this truncated geometric series is given by

$$1 + \sum_{i=0}^I \mathcal{P}_i = \mathcal{P}_{I+1} . \quad (5.29)$$

Examples of operations in the periodic-FMM are shown in Fig. 5.4, from level  $i = 0$  up to the illustrative maximum level  $I = 3$ . There, the red arrows indicate the results from M2M clustering operations, while the red dots are the center of the dark gray boxes to be clustered. The blue arrows represent the M2L operations from the center of the source-cluster, indicated by blue dots and light gray boxes. One must note that M2M operations are the same for both left and right sides of the central box since their computations depend only on the center of their upper boxes, so they are only performed once. However, the M2L operations must be performed twice to account for both left and right sides due to their different relative position with respect to the central domain. One must not forget that due to the separation criterion, the immediate neighbors  $|n| = 1$  of the central domain only perform M2L operations at levels  $\ell \geq 1$ . This way, their interactions are not shown in the figure below. Also, from Eq. 5.29, one can count  $2^{I+1} = 2^4 = 16$  baseline-boxes at level  $\ell = 0$ .

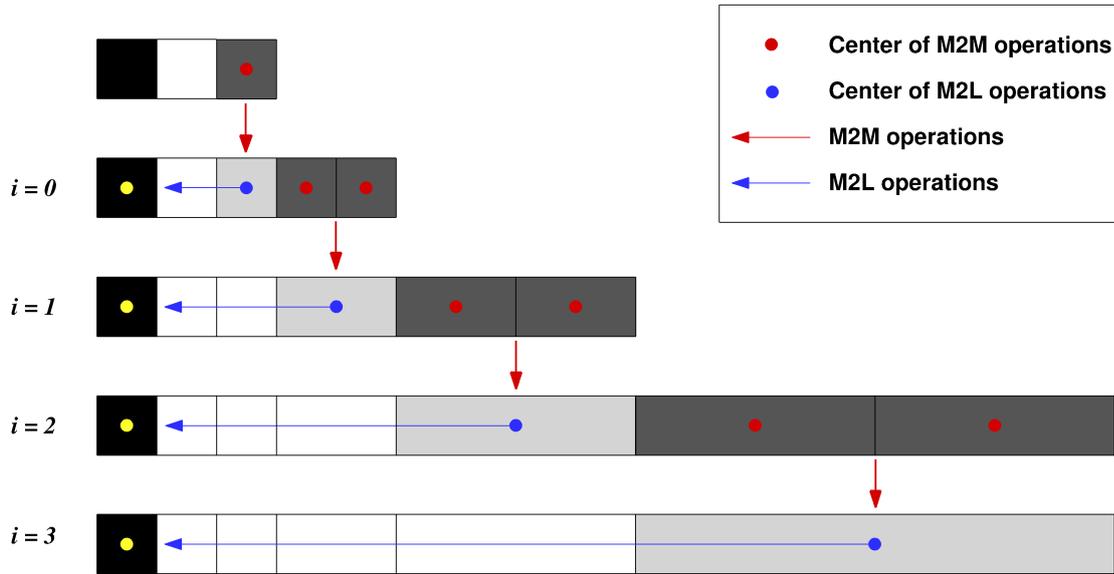


Figure 5.4: Optimized scheme for far-field interactions of periodic domains.

Here, the free-domain FMM formulation still remains and, therefore, the truncation term in the FMM series is  $p$  and the cluster influence is grouped into multipole expansions  $\mathcal{M}$  and local expansion  $\mathcal{L}$ . However, Eq. 3.21 is slightly changed to account for only 2 smaller boxes, instead of the 4 boxes used in the free-domain FMM. These 2 boxes from periodic level  $i - 1$  can be clustered at level  $i$ , for  $i \geq 1$ , using Eqs. 5.30 – 5.32:

$$\mathcal{M}_k^{(i)} = \sum_{m=1}^k \left[ \mathcal{M}_m^{(i-1)} \binom{k-1}{m-1} (z_{i-1,1}^{k-m} + z_{i-1,2}^{k-m}) \right] - \mathcal{M}_0^{(i-1)} \frac{(z_{i-1,1}^k + z_{i-1,2}^k)}{k}, \quad (5.30)$$

and the distances in the complex plane from the center of both left and right clusters to

the new cluster center are, respectively

$$z_{i-1,1} = -2^{i-2} \quad (5.31)$$

$$z_{i-1,2} = +2^{i-2} . \quad (5.32)$$

The periodic level  $i = 0$  coincides with the free-domain level  $\ell = 0$ , so  $\mathcal{M}_0^{(0)}$  is the same for both schemes. For levels  $i > 0$ , it is given by:

$$\mathcal{M}_0^{(i-1)} = 2^{i-1} \mathcal{M}_0^{(0)} . \quad (5.33)$$

After creating the larger clusters up to level  $I$ , in the periodical domains, their influences must be accounted directly at the central domain for all levels  $i = 0$  to  $I$ . They are evaluated using only the M2L step, with no further L2L operations. For  $m = 1$  to  $p$ , Eq. 3.28 is slightly changed to account for both left and right side domains, indicated by the subindex  $s$ . Finally, the local expansion can be written as

$$\mathcal{L}_m^{(M2L)} = \sum_{i=0}^I \sum_{s=1}^2 \left\{ -\frac{\mathcal{M}_0^{(i)}}{m \cdot z_{i,s}^m} + \frac{1}{z_{i,s}^m} \sum_{k=1}^p \left[ \mathcal{M}_k^{(i)} \binom{m+k-1}{k-1} \frac{(-1)^k}{z_{i,s}^k} \right] \right\} . \quad (5.34)$$

The distance from the left and right clusters to the central domain are, respectively,

$$z_{i,1} = + (1.5 \times 2^i + 0.5) \quad (5.35)$$

$$z_{i,2} = - (1.5 \times 2^i + 0.5) . \quad (5.36)$$

The scheme shown in Fig. 5.4 is performed to reduce the cost of operations at level  $\ell = 0$ , as indicated in Fig. 5.3(b). After these translations to the central domain, the computation proceeds for levels  $\ell > 0$ , as indicated by Fig. 5.3(c) and (d). The computational cost to perform one clustering at the periodic upward pass from level  $i - 1$  to  $i$  is  $O(p^2)$ , where  $p$  is the power series truncation term from the free-domain FMM. The multipole-to-local operation also has a cost  $O(p^2)$ . It is worth to note that this cost does not depend on the number of vortex particles in the problem. This way, for a maximum periodic level  $I$ , the total periodicity cost to account for all boxes is

$$O(p^2 \log_2(\mathcal{P}^{I+1})) . \quad (5.37)$$

If one employs the naive implementation of the scheme given by Eq. 5.25, the total cost should be orders of magnitude higher than that for the optimized scheme. The cost for a naive implementation is

$$O(p^2 \mathcal{P}^{I+1}) . \quad (5.38)$$

### 5.1.4 Desingularization

The usage of Eq. 5.10 leads to a singularity for the arguments  $z = -1, 0$  and  $1$ , *i.e.*, two vortex particles in close proximity to each other or to a periodic image. This way, the same mechanisms used in DVM to avoid the singularity can be employed for the periodical case.

One possibility is to use an array of Lamb-Oseen vortices, leading to

$$w_i = \sum_{n=-\infty}^{\infty} i \frac{\Gamma_j}{2\pi} \left( \frac{1}{z_{ij} + n} \right) \left\{ 1 - \exp \left[ -\frac{(x_{ij} + n)^2 + (y_{ij})^2}{\sigma^2} \right] \right\}, \quad (5.39)$$

and one can rewrite the induced velocity as

$$w_i = i \frac{\Gamma_j}{2\pi} \left\{ \sum_{n=-\infty}^{\infty} \left( \frac{1}{z_{ij} + n} \right) - \sum_{n=-\infty}^{\infty} \left( \frac{1}{z_{ij} + n} \right) \exp \left[ -\frac{(x_{ij} + n)^2 + (y_{ij})^2}{\sigma^2} \right] \right\}, \quad (5.40)$$

where the first term is the cotangent. The second term has exponential decay and, for  $|n| > 1$  and  $\sigma^2 \ll 1$ , it vanishes. Hence, a simplified expression is written as

$$w_i = i \frac{\Gamma_j}{2\pi} \left\{ \pi \cot(\pi z_{ij}) - \sum_{n=-1}^1 \left( \frac{1}{z_{ij} + n} \right) \exp \left[ -\frac{(x_{ij} + n)^2 + (y_{ij})^2}{\sigma^2} \right] \right\}, \quad (5.41)$$

where a vortex in the central domain, and its images at  $|n| = 1$ , are computed as Lamb-Oseen vortices and those distant from the central domain are computed using the potential vortex model through the FMM.

The numerical desingularization from Krasny (1986a) can be applied in order to modify Eqs. 5.17 and 5.18. This leads to

$$u_i = \frac{\Gamma_j}{2\pi} \left( \frac{+ \pi \sinh(2\pi y_{ij})}{\cosh(2\pi y_{ij}) + \cos(2\pi x_{ij}) + \delta^2} \right), \quad (5.42)$$

$$v_i = \frac{\Gamma_j}{2\pi} \left( \frac{- \pi \sin(2\pi x_{ij})}{\cosh(2\pi y_{ij}) + \cos(2\pi x_{ij}) + \delta^2} \right). \quad (5.43)$$

The distances  $x_{ij}$  along the  $x$  axis and  $y_{ij}$  along the  $y$  axis are given respectively by Eqs. 5.19 and 5.20.

## 5.2 Results

This section shows a comparison of both non-singular vortex models presented above, including their velocity profile and the long-time integrations of the roll-up of a shear-layer using both models. Also, the velocity field for the vortex sheet roll-up is calculated by Biot-Savart law with cotangent function and a periodic implementation of the FMM, with additional terms to avoid a spurious precession of the shear-layer. Finally, numerical perturbations lead to a wrong solution, where an assessment of best practices to avoid this spurious instabilities is performed.

### 5.2.1 Velocity profile

The tangential velocity profile was calculated for three models, Eqs. 5.10, 5.43 and 5.41. The relative deviation for Blob and Lamb-Oseen vortex models compared to the exact solution is given by

$$\epsilon_{model} = \frac{w_{cot} - w_{model}}{w_{cot}} , \quad (5.44)$$

while the absolute error

$$E_{model} = w_{cot} - w_{model} . \quad (5.45)$$

The vortex blob induces higher velocity in close distance, while the Lamb is smoother, as shown in the auto similar velocity profile in Fig. 5.5. For small distances,  $z \rightarrow 0$ , the cotangent solution is nearly the same for a single potential vortex. The Lamb-Oseen model converges exponentially to the solution, while the blob model converges in an algebraic sense, as shown in Figs. 5.6 and 5.7. Again, only the first one can be used with the FMM without loss of precision.

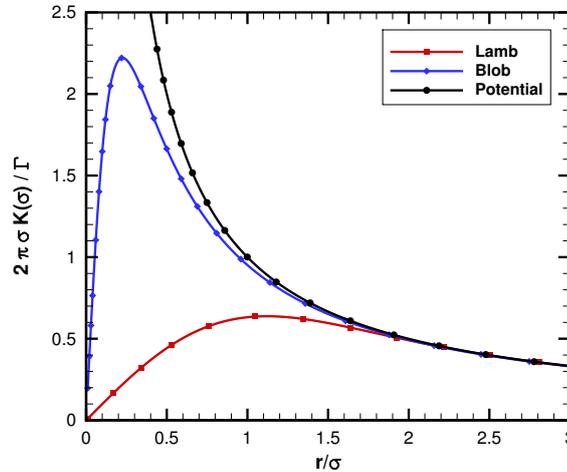
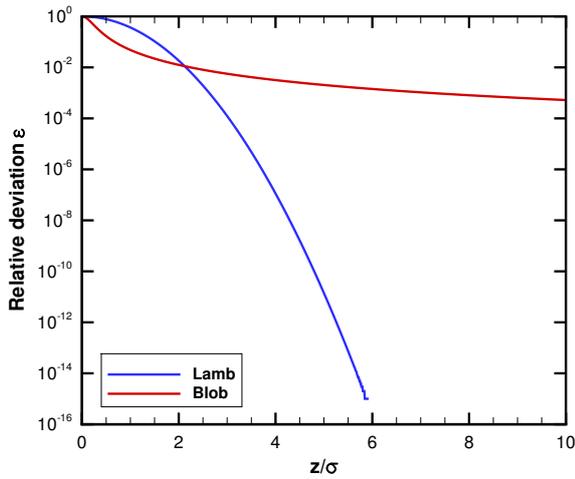
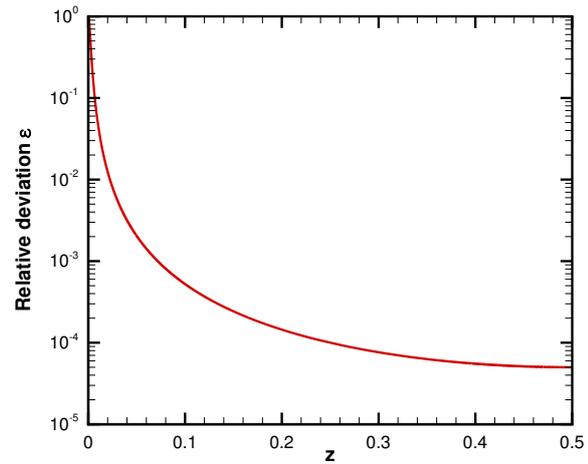


Figure 5.5: Velocity profile for periodic vortex models.

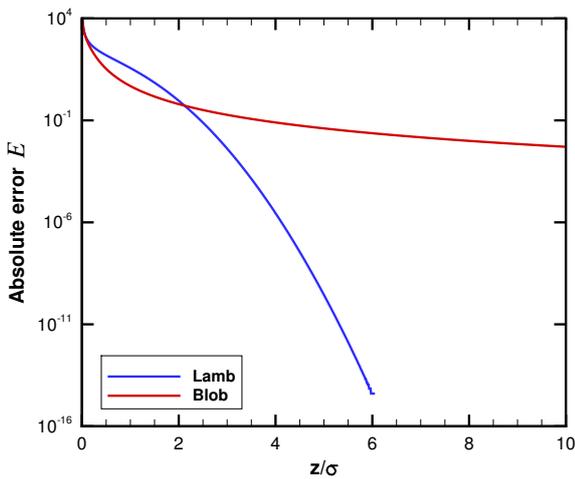


(a) Comparison of error for both models.

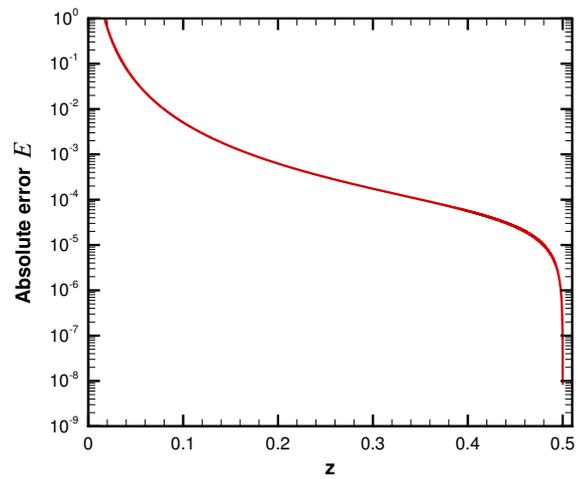


(b) Vortex blob's error decay.

Figure 5.6: Relative error for both periodic vortex models.



(a) Comparison of error for both models.



(b) Vortex blob's error decay.

Figure 5.7: Absolute error for both periodic vortex models.

## 5.2.2 Biot-Savart solution

A comparison of the solution of both vortex models is illustrated in Fig. 5.8, where it is possible to see that vortex blob rolls-up faster than Lamb, leading to a higher number of spirals in the center due to the more singular shape of the velocity profile, as shown before in Fig 5.5. The shear layer is discretized by 5120 vortex with  $\sigma = \delta = 0.05$ , advected using fourth-order Runge-Kutta time-marching scheme (RK4) and  $\Delta t = 0.01$  up to  $t^* = 2.0$ . The sinusoidal initial amplitude  $A$  is equal to 0.01.

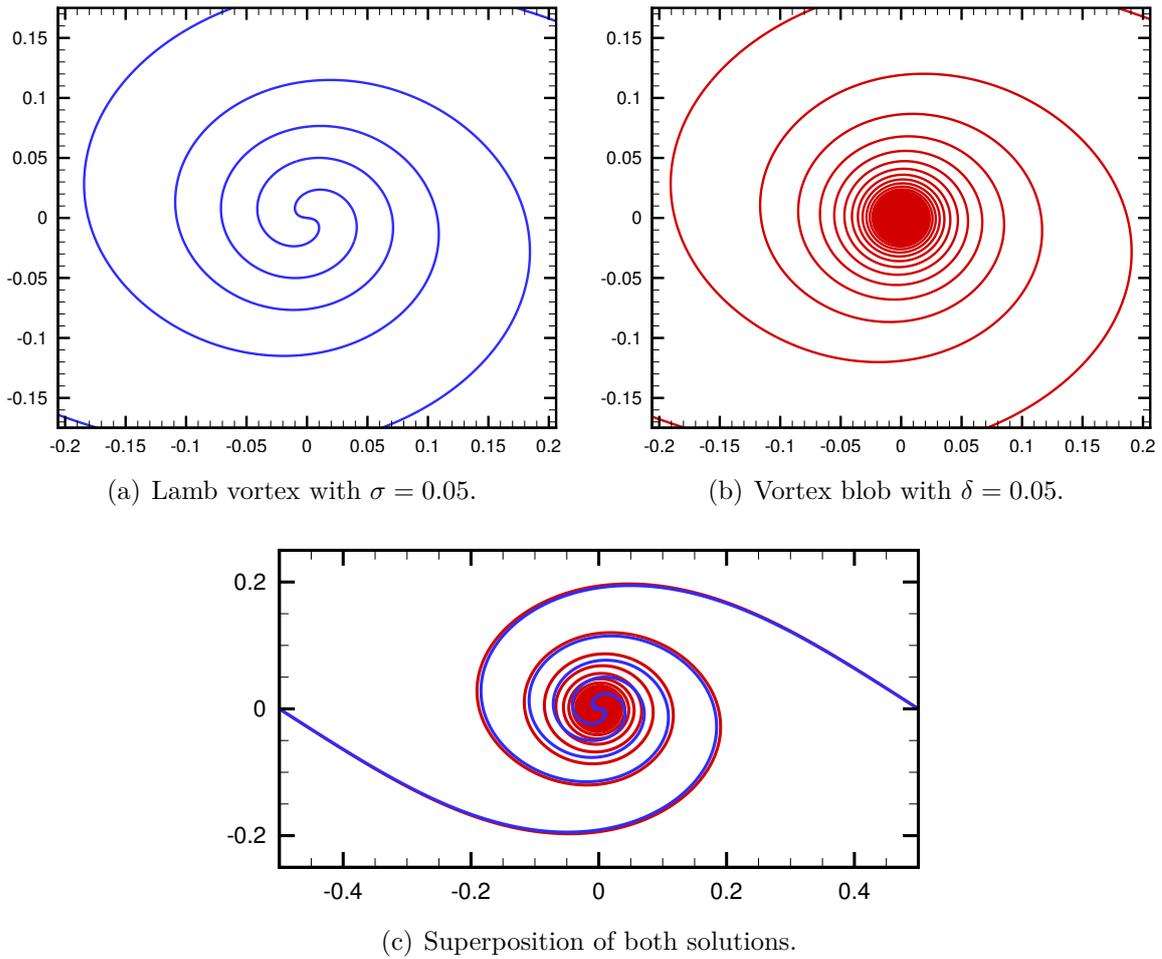


Figure 5.8: Results for Biot-Savart with  $\sigma = \delta = 0.05$ , 5120 vortex, RK4,  $\Delta t = 0.01$ ,  $t^* = 2.0$ , and  $A = 0.01$ .

In absence of the Hamiltonian to check for invariants on convection for a Lamb-Oseen vortex array system, a quick investigation was performed to determine a good value of time-step, which allows a fast simulation without compromising the error. The deviation in position of the vortices in  $t^* = 2.0$  is calculated using Eq. 5.46 for different values of time steps  $\Delta t$  is compared to a time-step of  $\Delta t = 0.001$ . This value is the smallest used by Krasny (1986a,b). Also, the tests were performed using viscous core given by  $\sigma = 0.02$  and  $0.05$

$$\Delta \mathbf{x} = \mathbf{x}_{0.001} - \mathbf{x}_{\Delta t} \quad (5.46)$$

The deviation in position is shown in table 5.1, while the vortices can be seen in Fig. 5.9, where a convergence to a solution is present. The black dots in all three sub-figures are the solution for the smallest time-step investigated,  $\Delta t = 0.001$ , while the colored lines are the solution for the time-step indicated in the captions. A time step  $\Delta t = 0.01$  grants small  $l_\infty$  norm error at  $t^* = 2.0$  without being necessary to calculate lots of steps, so this value is used for all computations from here.

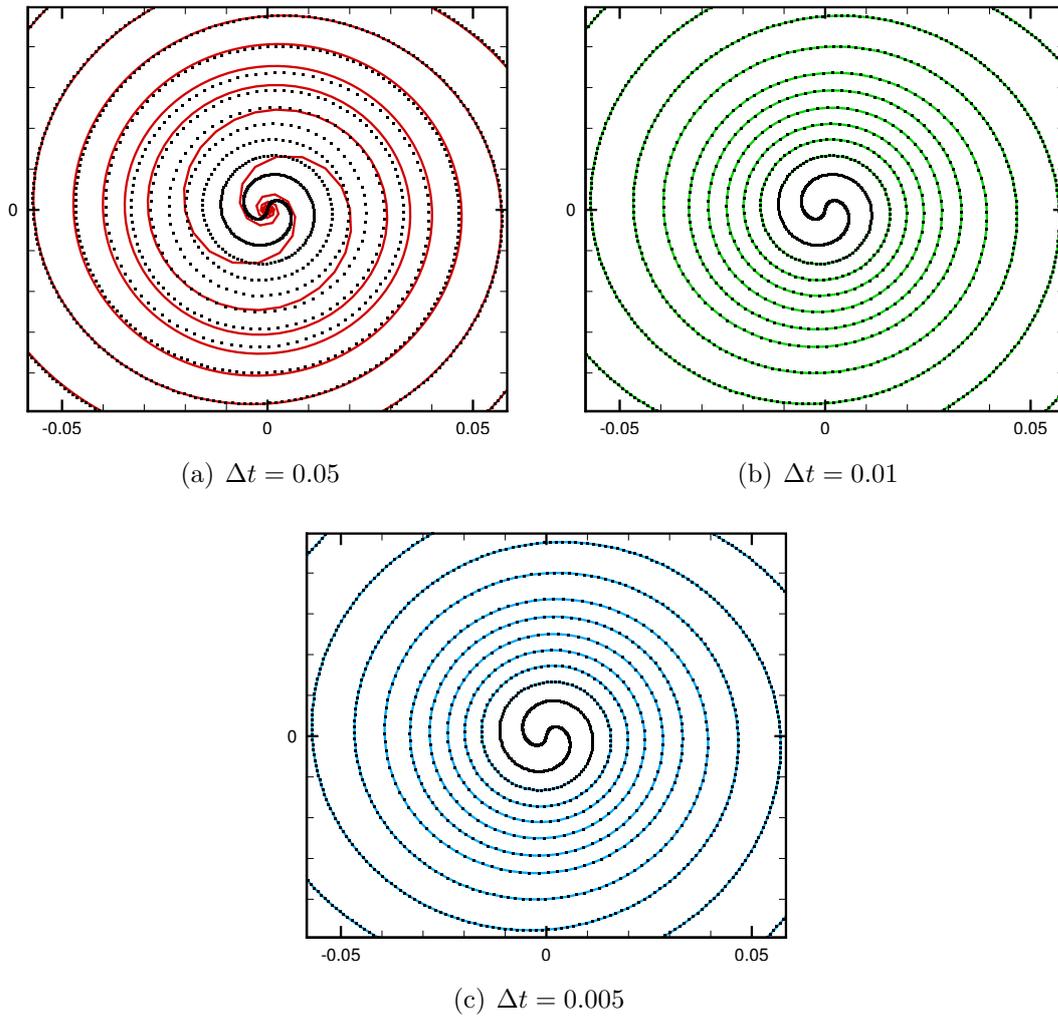


Figure 5.9: Convergence study for the position of 5120 vortex with  $\sigma = 0.02$  at  $t^* = 2.0$  for different time steps.

Table 5.1: Accumulated error for different time steps, compared to  $\Delta t = 0.001$ , to march the system to  $t^* = 2.0$

Viscous core	Time step	RMS norm	$l_\infty$ norm
$\sigma = 0.05$	0.1	1.556e-3	4.597e-3
	0.05	5.670e-5	1.502e-4
	0.01	1.569e-7	3.930e-7
	0.005	1.070e-8	2.667e-8
$\sigma = 0.02$	0.1	3.338e-2	0.104
	0.05	6.864e-3	4.700e-2
	0.01	1.791e-5	8.302e-5
	0.005	3.306e-7	1.210e-6

If one reduces the viscous core going towards an inviscid solution, this absence of diffusion may trigger spurious perturbations on the sheet, leading to a wrong solution, as shown in Fig. 5.10 for  $\sigma = 0.01$ . It was investigated the possibility to increase the number of particles and then eliminate this perturbations, following the same procedure from chapter 4. However, the opposite was found and the solution is even worse when the number of elements increases. This is in accordance with the description by Krasny (1986b).

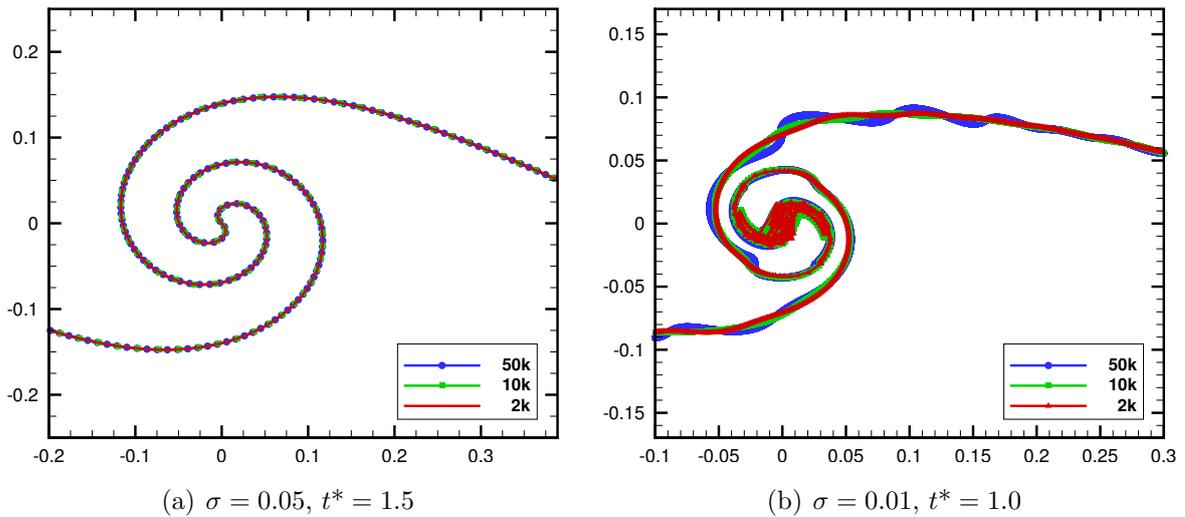


Figure 5.10: Effect of number of vortex using Biot-Savart with RK4,  $\Delta t = 0.01$ , and  $A = 0.01$ .

More tests were performed for 2k vortex with core  $\sigma = 0.01$  using Biot-Savart with RK4,  $\Delta t = 0.01$ , and  $A = 0.01$ . It is shown in Fig. 5.11(a) that this spurious perturbation begins at the center in  $t^* = 0.8$  and propagates, disrupting the entire solution at  $t^* = 1.2$ , Fig 5.11(b). In order to avoid this perturbation, the amplitude  $A$  of the initial sinusoidal displacement is reduced. Using an amplitude  $A = 0.005$ , it is seen that the central instability vanishes  $t^* = 0.8$ , Fig. 5.11(c), however the solution is even worse since simultaneous spurious perturbations with vortex pairing appears in the entire wake, Fig. 5.11(d) for  $t^* = 1.2$ .

Back to the initial amplitude  $A = 0.01$ , the usage of quadruple machine precision (16 bytes) in order to reduce truncation error was investigated. The core spurious perturbation is the same as before with double precision (8 bytes) at  $t^* = 0.8$ , but somehow the instabilities are not propagated, Fig. 5.12(a). For  $t^* = 1.2$ , the wake continues smooth everywhere except in the core region, Fig. 5.12(b). This shows that the core instability is associated to the larger initial amplitude, while the other instabilities are associated to absence of diffusion in formulation, which are quickly amplified with reduced machine precision.

Increasing the machine precision together with smaller amplitude, it was possible to obtain a smooth solution everywhere, as shown in Fig. 5.13(a-c) even for  $t^* = 2.0$ . The reduced number of elements leads to a low-resolution roll-up of the core, Fig. 5.13(d). Increasing the number of vortex from 2k to 10k, a smoother solution in the core is obtained, as shown in Fig. 5.13(e) and (f). It is also worth to note that the spurious perturbations does not appear for 10k vortex particles, on the contrary of Fig. 5.10. There, the solution for 10k vortex shows perturbations with higher amplitude than 2k particles.

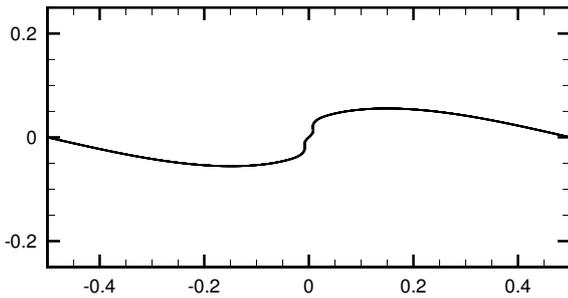
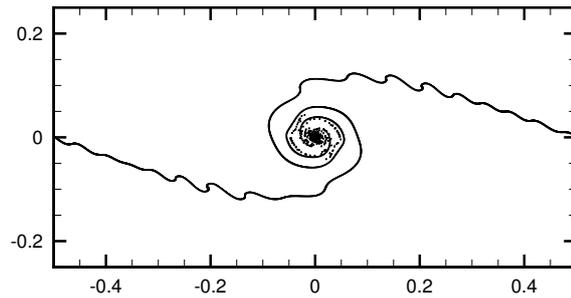
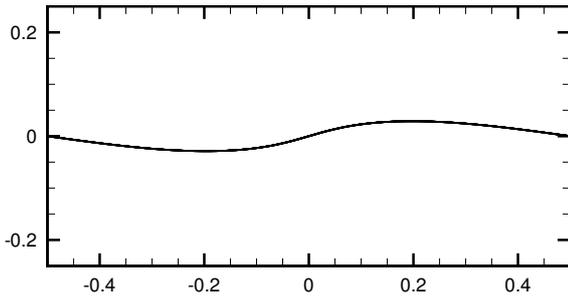
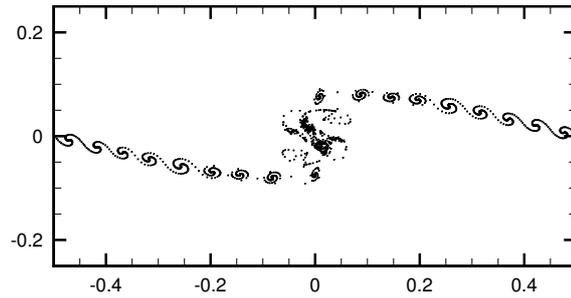
(a) Double precision with  $A = 0.01$  at  $t^* = 0.8$ (b) Double precision with  $A = 0.01$  at  $t^* = 1.2$ (c) Double precision with  $A = 0.005$  at  $t^* = 0.8$ (d) Double precision with  $A = 0.005$  at  $t^* = 1.2$ 

Figure 5.11: Effect of initial perturbation for 2k vortex with  $\sigma = 0.01$  using Biot-Savart, RK4 and  $\Delta t = 0.01$ .

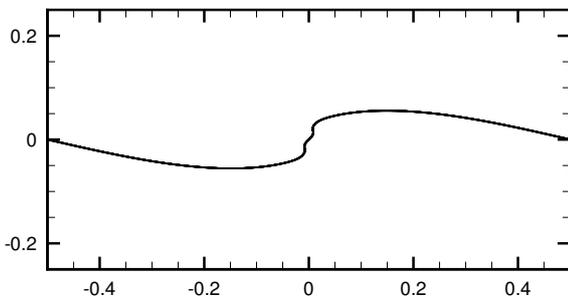
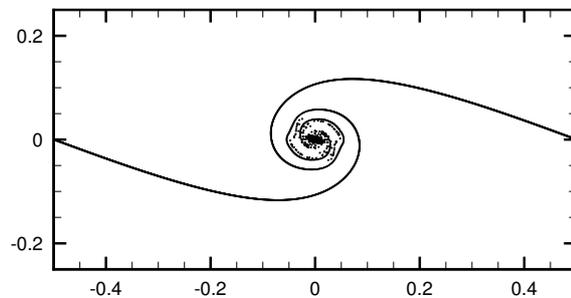
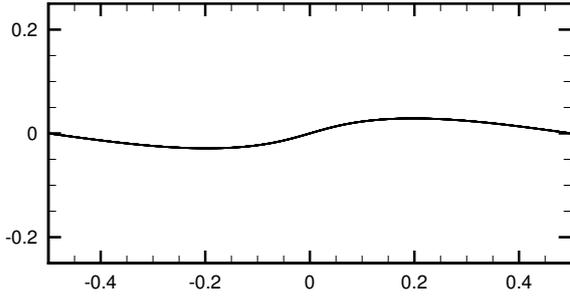
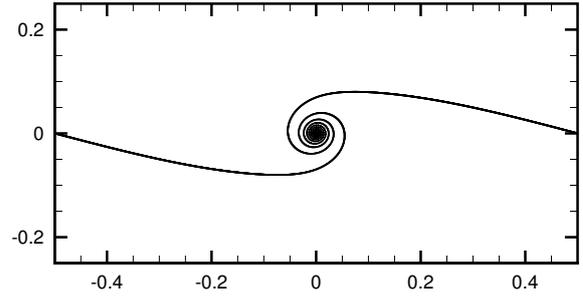
(a) Quadruple precision at  $t^* = 0.8$ (b) Quadruple precision at  $t^* = 1.2$ 

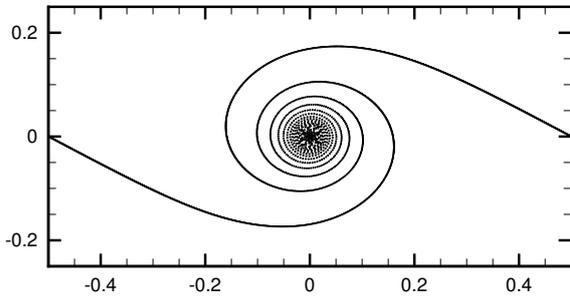
Figure 5.12: Effect of machine precision for 2k vortex with  $\sigma = 0.01$ ,  $A = 0.01$ , Biot-Savart, RK4 and  $\Delta t = 0.01$ .



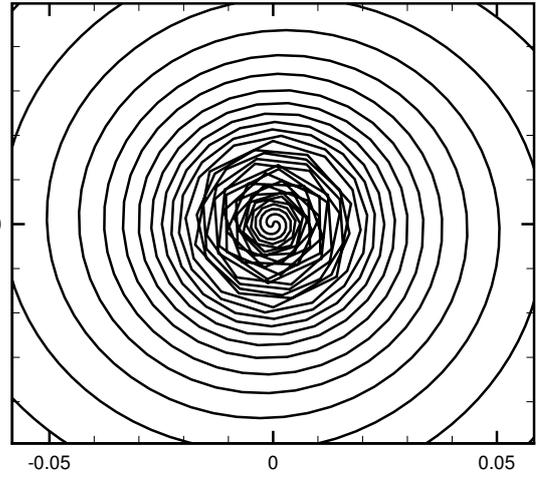
(a) 2k vortex, 16 bytes,  $A = 0.005$ ,  $t^* = 0.8$



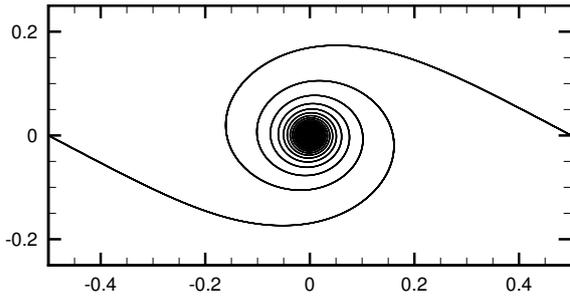
(b) 2k vortex, 16 bytes,  $A = 0.005$ ,  $t^* = 1.2$



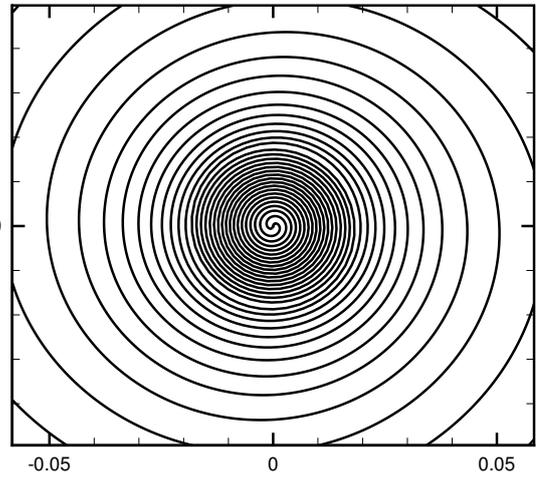
(c) 2k vortex, 16 bytes,  $A = 0.005$ ,  $t^* = 2.0$



(d) Detail on the core with low resolution.



(e) 10k vortex, 16 bytes,  $A = 0.005$ ,  $t^* = 2.0$



(f) Detail on the core with better resolution.

Figure 5.13: Combination of smaller amplitude  $A = 0.005$  and increased machine precision (16 bytes) with  $\sigma = 0.01$ , Biot-Savart, RK4 and  $\Delta t = 0.01$ .

### 5.2.3 Periodic FMM's convergence

The FMM is applied to accelerate the solution of periodic shear layer. The method is based on a series expansion of the cotangent function, which governs the vortex dynamics, and it may be truncated in  $p$  terms. The convergence study in Fig. 4.9 shows that machine precision is achieved for about  $p = 40$  terms, so this value is used in all simulations using the FMM in this chapter.

Also, both Eqs. 5.30 and 5.34 for FMM operations are results of power series manipulations dependent on the distance from a cluster of source to another with observers. When periodicity is applied, the distance from a cluster of images to the central domain is proportional to the periodic level  $i$ :

$$z \sim O\left(2^i\right), \quad (5.47)$$

so, a power series with exponents  $n = 1$  to  $p$  is written as:

$$z^n \sim O\left(2^{ni}\right), \quad (5.48)$$

which may be close to the machine upper limit precision depending on both  $n$  and  $i$ . So, to avoid *NaN* in computations, the truncation of the power series may occur with less terms than  $p$  for very distant clusters, based on the value of the  $z^n$  term.

Other source of error comes from the replication of domains in FMM, and it arises from the truncation of an infinite shear layer. Each domain represents a small shear layer that works as a boundary condition for the central domain. The shear layers are not coherent structures, so their tips are free to move. For a distribution of  $N$  vortex particles, there is an asymmetric distribution of elements around an observer  $m$ , shown in Fig. 5.14. This unbalance induces a higher vertical velocity in  $m$ , and it is expected to be higher if  $m$  is further from the center.

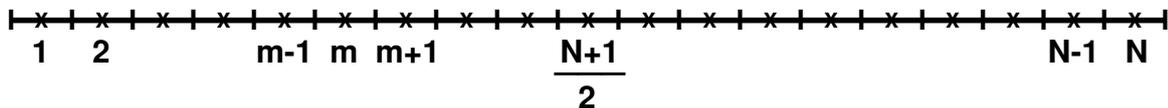


Figure 5.14: Discretization of a finite shear layer using  $N$  vortex elements.

The induced velocity moves the particles such that the wake tips begins to roll-up. Also, the entire wake have a precession movement about its center  $x_c$ , placed at:

$$x_c = \frac{N + 1}{2}. \quad (5.49)$$

From the induced velocity by a potential vortex at left of  $m$ :

$$v_{left} = \sum_{i=1}^{m-1} \frac{1}{N} \frac{1}{\Delta x_i}, \quad (5.50)$$

while by those at the right side is:

$$v_{right} = \sum_{i=m+1}^N \frac{1}{N} \frac{1}{\Delta x_i}, \quad (5.51)$$

with the vortex separation given by:

$$\Delta x_i = x_i - x_m. \quad (5.52)$$

Then, combining Eq. 5.50 and a expanded Eq. 5.51, it is possible to write the induced velocity  $v$  at  $m$ :

$$v_m = \sum_{i=1}^{m-1} \frac{1}{N} \frac{1}{\Delta x_i} + \sum_{i=m+1}^{2m-1} \frac{1}{N} \frac{1}{\Delta x_i} + \sum_{i=2m}^N \frac{1}{N} \frac{1}{\Delta x_i}, \quad (5.53)$$

where, by symmetry, the first two terms cancel each other. This way, the induced velocity is equal to the remaining term:

$$v_m = \sum_{i=2m}^N \frac{1}{N} \frac{1}{\Delta x_i}, \quad (5.54)$$

and it is only null if  $2m - 1 = N$ , *i.e.*, at the center of rotation given by Eq. 5.49.

Adding  $n$  domains to both tips, *i.e.* going from a free-domain layer to a periodic layer, the asymmetry of charges are moved further away. If  $n \gg \Delta x_i$ , the following approximation is valid:

$$v_{left} = \sum_{i=1}^{m-1} \frac{1}{N} \frac{1}{\Delta x_i - n} \approx \frac{m-1}{N} \frac{1}{\Delta x' - n}, \quad (5.55)$$

where  $\Delta x'$  is the mean separation of the elements at left. The same is valid for the right side elements, where the mean separation is  $\Delta x''$ :

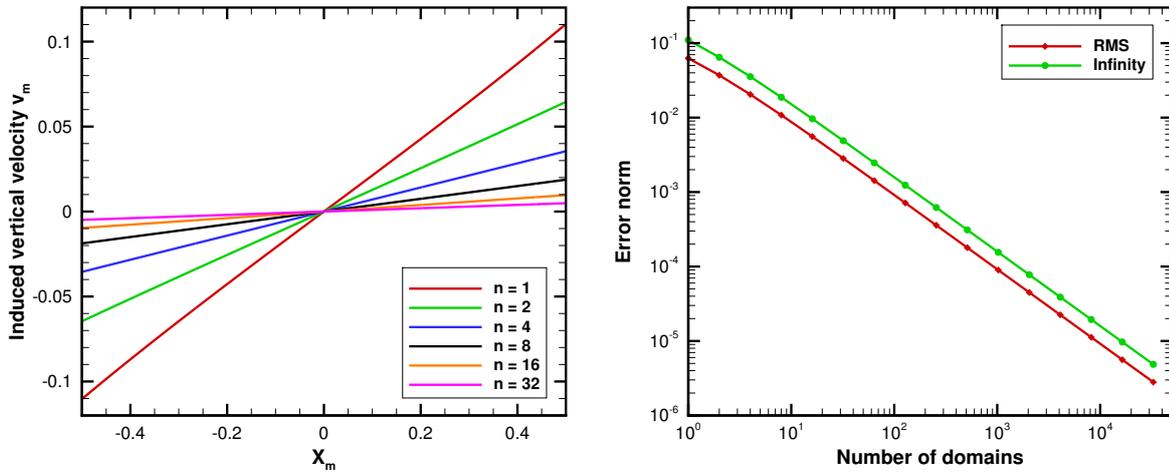
$$v_{right} = \sum_{i=m+1}^N \frac{1}{N} \frac{1}{\Delta x_i - n} \approx \frac{N-m}{N} \frac{1}{\Delta x'' + n}, \quad (5.56)$$

and the summation of both equations leads to the reduced induced precession velocity. Since  $n \gg \Delta x$ , further simplifications can be done and the induced velocity  $v_m$  is given

by Eq. 5.57:

$$v_m \sim O\left(\frac{2m - N - 1}{N} \frac{1}{n}\right). \quad (5.57)$$

The measurements in Fig. 5.15 are performed for 5120 vortex particles in the first time step of a straight shear layer, where it is expected from a delicate balance that the induced vertical velocity is null, as described by Lewis (1991). It confirms that a vortex close to the center has smaller precession speed while the tip is faster, as shown in Fig. 5.15(a). Also, the error decay based on the number of domains is  $O(1/n)$  and slowly converges, as shown in Fig. 5.15(b).



(a) Vertical velocity based on the number  $n$  of periodical domains.

(b) Error from the expected velocity.

Figure 5.15: Precession velocity for a shear layer discretized with 5120 vortex particles according to the number of periodic domains.

In the case of an initially perturbed shear layer with sinusoidal shape, the problem is a little more complex. The results for long-time integrations are shown in Fig. 5.16, where the use of additional domains reduce both precession and secondary roll-up. It is possible to see a convergence to the solution of the Biot-Savart law using the cotangent function. The scheme employed is given by Eq. 5.25 to march 5120 vortex with viscous core  $\sigma = 0.05$  using the RK4 scheme and  $\Delta t = 0.01$  until  $t^* = 2.0$ . The initial amplitude of the sinusoidal perturbation is  $A = 0.01$ .

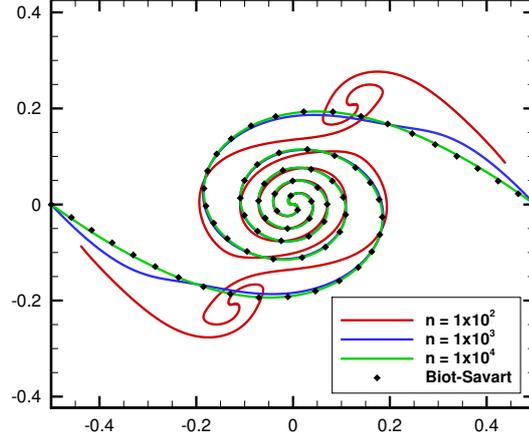


Figure 5.16: Influence of number of domains in periodicity for a finite shear layer discretized by 5120 vortex particles,  $A = 0.01$ ,  $\sigma = 0.05$ , RK4 with  $\Delta t = 0.01$  until  $t^* = 2.0$ .

Since this precession is purely due to the unbalance from truncating a periodic shear layer, one must overcome its presence. If  $N - m$  charges with circulation  $\Gamma_i = 1/N$  are placed at the left side of a vortex  $m$  in the  $n + 1$  domain, *i.e.*, after the last domain  $n$ , there is a induced velocity:

$$v_{left} = \sum_{i=m+1}^N \frac{1}{N} \frac{1}{\Delta x_i - (n+1)}, \quad (5.58)$$

and another set of  $m - 1$  charges to the right of the vortex:

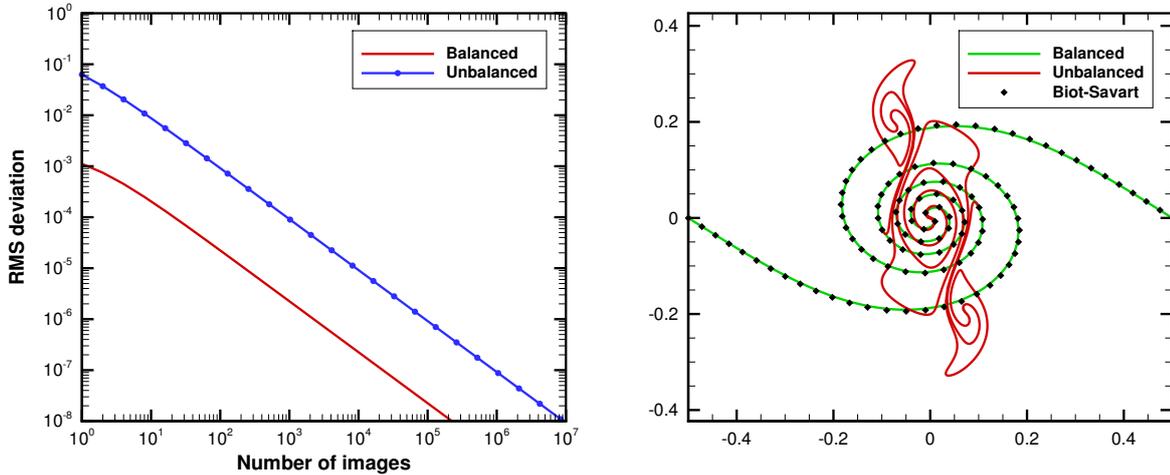
$$v_{right} = \sum_{i=1}^{m-1} \frac{1}{N} \frac{1}{\Delta x_i + (n+1)}, \quad (5.59)$$

Combining Eqs. 5.50, 5.51, 5.58 and 5.59, one will have exactly  $N$  particles after the vortex  $m$ . By symmetry, these additional particles cancel each other so there is not a precession speed about  $m$ . This simple balancing of charges eliminates any precession in a horizontal, straight shear layer since all vortex have null velocity, and any error is purely due to the machine precision.

However, a perturbed layer, Eq. 5.5, no longer has symmetry inside the central domain and it also has the  $u$  component of the velocity. This way, compared to the cotangent solution, there is still an error for induced velocity proportional to the number of periodic images at both sides of central domain. Figure 5.17(a) presents the RMS deviation calculated with 5120 vortex particles only for the first step of simulation. It is possible to see that the proposed balancing reduces the error and accelerates the convergence.

The use of shear layer balancing is more than just error-reducing. In Fig. 5.17(b), for only 10 domains at both sides of the central domain, the balanced result is very close to

the Biot-Savart law solution using Lamb-Oseen vortex array, Eq. 5.41. The unbalanced solution not only has precession but also secondary rolls-up, leading to a wrong solution. The results are obtained from a simulation with a shear layer discretized by 5120 vortex,  $A = 0.01$ ,  $\sigma = 0.05$ , RK4 with  $\Delta t = 0.01$  for  $t^* = 2.0$ .

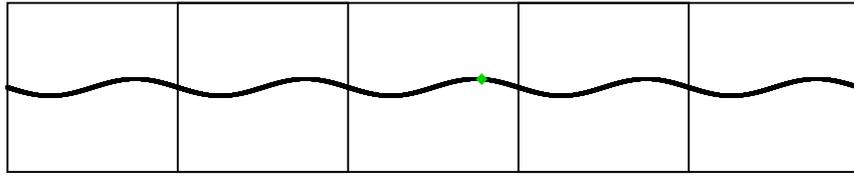


(a) Rate of convergence for the periodic FMM with and without balancing. (b) Influence of balance usage for 10 periodic domains representing a finite shear layer.

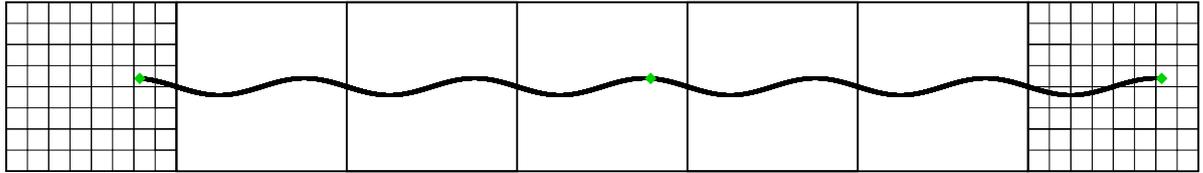
Figure 5.17: Error reduction when using a balanced shear layer.

Figure 5.18(a) presents an unbalanced shear layer. Since balancing is necessary, the FMM can be further modified to consider the additional particles and to avoid excessive computational cost. The whole balancing procedure is explained in Fig. 5.18. The main idea is to cluster, at the left, the particles indicated by  $k + 1$  to  $N$ , while at the right side, the cluster is from 1 to  $k - 1$  (see Fig. 5.18(b)). The clustering uses the existent multipole expansions from the free-domain FMM for levels  $\ell = 1$  to  $L - 1$ . The boxes which are clustered are those in the inner portion of the wake shown in light gray in Fig. 5.18(c). If any of the white boxes appearing in this previous figure is used, unbalancing will occur.

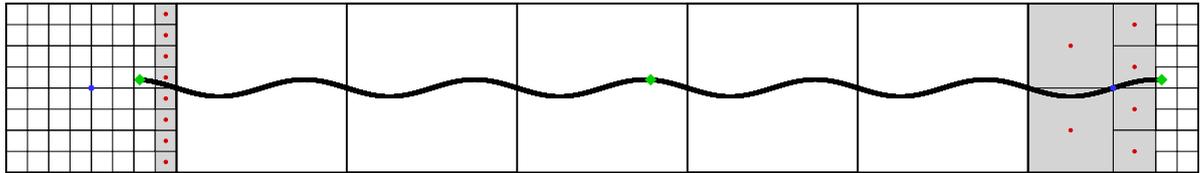
From the clustering, two new equivalent clusters at the boundary of the truncated domain are created as presented in Fig. 5.18(d). One cluster is at the left and another at the right side. These two new clusters interact directly with the smallest box (dark gray in the figure), in the central domain, at level  $L$ , using M2L operations with no further L2L operations. For a precise balancing, the particles inside both replicated finer boxes in the boundaries, at level  $L$ , interact with the dark gray boxes directly using the Biot-Savart law. Hence, a new  $O(\tilde{N}^2)$  cost is added, where  $\tilde{N}$  is the number of particles in the column of finest level boxes. For a homogeneous distribution of elements, this new cost may be significant, while for heterogeneous distributions, this cost can be small based on the spatial distribution of vortex particles.



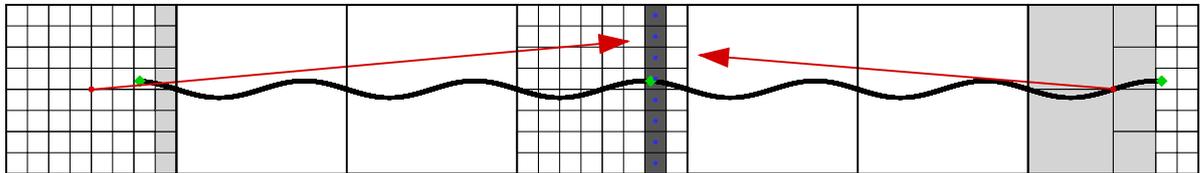
(a) Periodicity of the FMM without balancing. Here, the periodicity is considered with respect to the green vortex particle in the central domain.



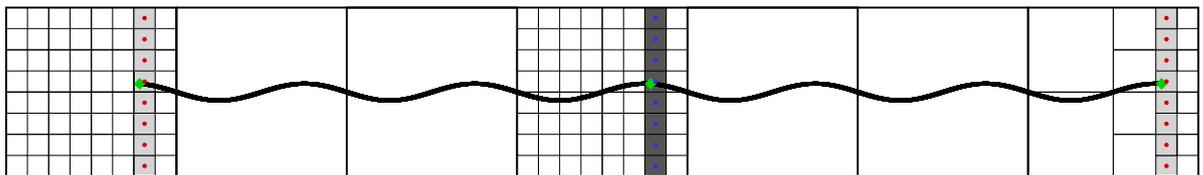
(b) Creation of boxes at the finest level in both boundaries of the periodicity.



(c) M2M operations to level 0 of the boxes in the internal part of the wake (red dots to blue dots).



(d) M2L operations using the multipole expansions at both tips of the wake (red dots to blue dots in dark gray boxes).



(e) Direct interactions in the smaller boxes coincident with the replica of the main particle (particles inside boxes with red dots to green particle in the central domain).

Figure 5.18: Current implementation of the wake balance procedure using the FMM.

The convergence based on the number of domains,  $n$ , of the solution with balancing is investigated for long-time integrations. The computation of the FMM periodicity uses the optimized fast summation of the far-field using Eqs. 5.26 – 5.36. The solutions converge to a similar shear layer pattern that does not depend on the number of domains employed. The calculations are performed with 5120 vortex particles,  $\sigma = 0.02$ ,  $A = 0.005$ , RK4,  $\Delta t = 0.01$  for  $t^* = 2.0$ , and the results are shown in Fig. 5.19.

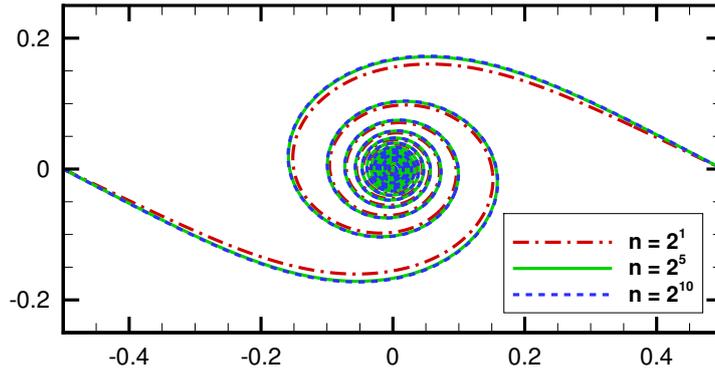


Figure 5.19: Investigation of the number of domains in periodic FMM using 5120 vortex,  $A = 0.005$ ,  $\sigma = 0.02$ , RK4 with  $\Delta t = 0.01$  until  $t^* = 2.0$ .

From the results shown in Fig. 5.19, it is possible to measure the long-time integration error of the periodic FMM compared to the exact cotangent solution in different time instants  $t^*$ . For short time integration, the error decays as  $O(1/n)$ , where  $n$  is the number of images. However, for longer time integrations, the cumulative error creates a plateau on the graph, independent on the number of domains. One can see these results in terms of both RMS deviation and  $l_\infty$  norm in Figs. 5.20(a) and (b), respectively.

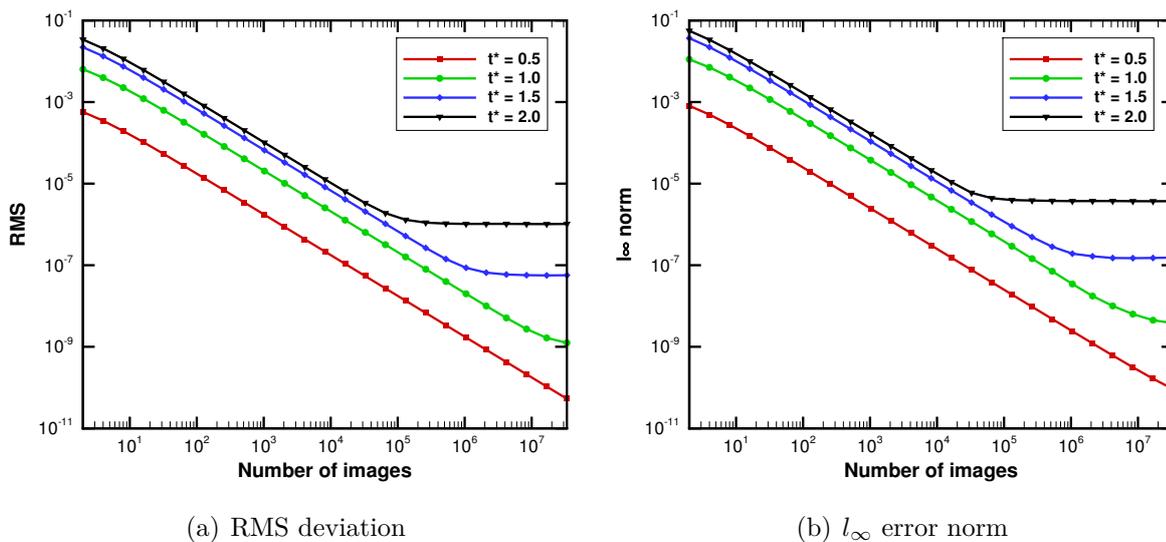


Figure 5.20: Long-time integration error for 5120 vortex,  $A = 0.005$ ,  $\sigma = 0.02$ , RK4 with  $\Delta t = 0.01$  until  $t^* = 2.0$ .

The direct solution using Biot-Savart law presents spurious instabilities similar to those from the Trefftz plane when  $\sigma \rightarrow 0$ . The issues are solved using smaller sinusoidal displacements for the initial condition as well as quadruple precision for the computations. Hence, the FMM is applied to solve this reduced-viscosity problem using 50k discrete vortex particles with  $\sigma = 0.01$ . The results are shown in Fig. 5.21 for different time steps, indicating the time evolution of the periodic shear-layer. Moreover, only the central domain is indeed solved, *i.e.*, the region from  $-0.5$  to  $+0.5$ . The region  $|x| > 0.5$  are merely replicas of the central domain.

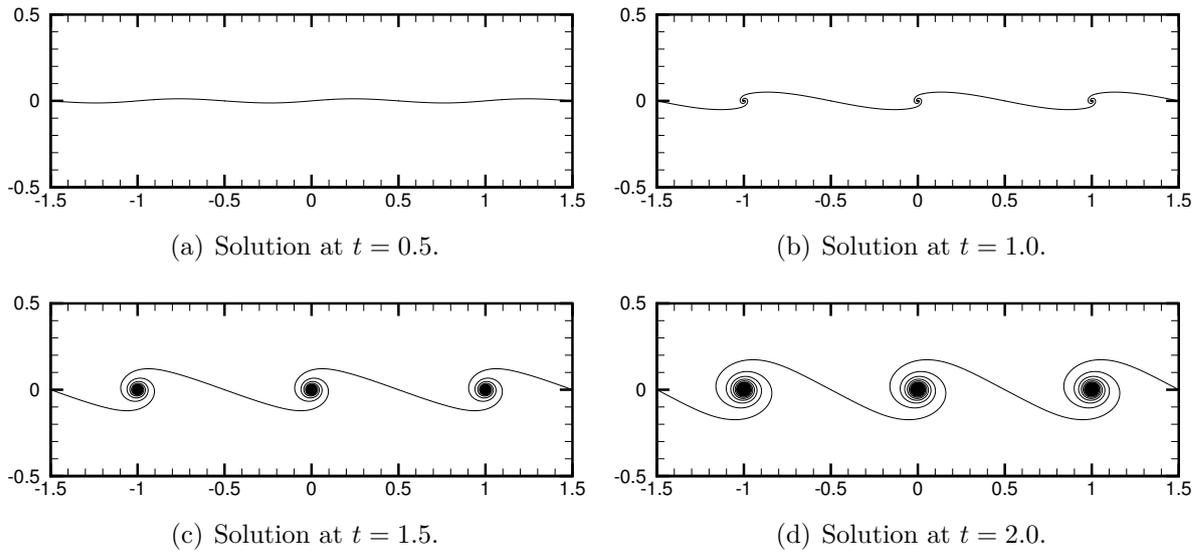


Figure 5.21: Temporal evolution of shear-layer discretized by 50k elements with  $\sigma = 0.01$  convected by RK4 with  $\Delta t = 0.01$  to  $t^* = 2.0$  using periodic FMM and quadruple precision.

One can see that the solution is smooth without any spurious instabilities, on the contrary of the results shown in Fig. 5.10 for the same number of vortex particles. A detail view of the core region is shown in Fig. 5.22, showing several layers of vorticity.

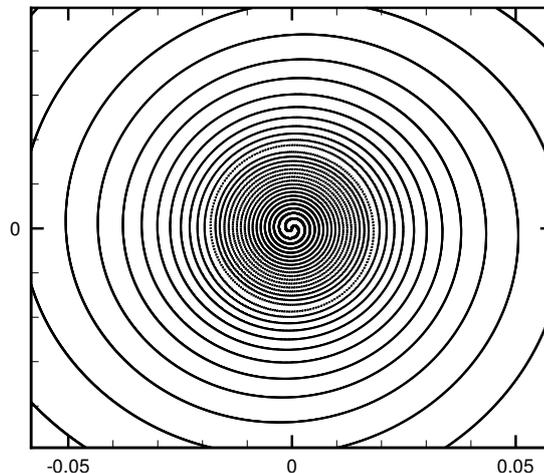
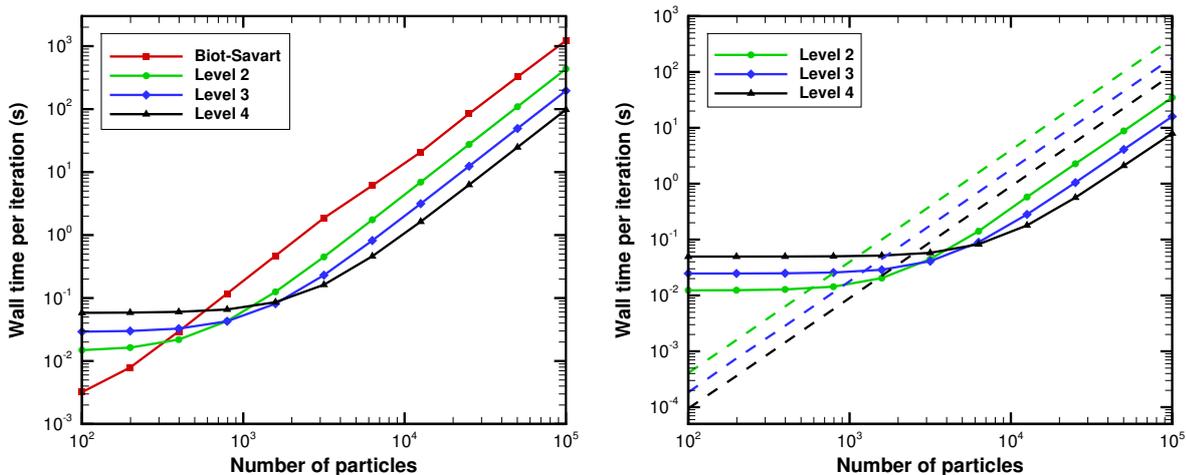


Figure 5.22: Detail view on the core region for long-time integration.

## 5.2.4 Computational Time

The computational time of the periodic FMM is investigated using a 2.5 GHz Intel® Xeon™ E5-2670v2 CPU in serial calculations. Besides from the total computational cost, two new costs must be analyzed: the cost from balancing as well as the level 0 periodicity to account for the  $n$  periodic boxes. A comparison of total cost from direct evaluation using Biot-Savart cotangent solution with the periodic FMM was performed up to  $1 \times 10^5$  elements and the results are shown in Fig. 5.23(a). The fixed cost is relevant for small number of particles, however, the time reduction using FMM is about 10 times faster than direct summation for the maximum number of particles investigated. Greater cost reductions could be achieved for finer levels of refinement in the FMM but the limitation on the relation between viscous core and the box size limits the savings. In comparison with Fig. 4.22, the total cost from both free-domain and periodic FMM are very similar considering the same levels of refinement.

The balancing cost of the periodic FMM is shown in Fig. 5.23(b), where the solid lines represent the combination of both M2L and P2P operations involved in the balancing procedure (Eqs. 5.58 and 5.59). Also, this new cost is compared to the near-field computations inside the central domain, shown in dashed lines. Although a new  $O(N^2)$  cost is added, this operation is nearly 10% of the existing near-field cost for large and it does not compromise the use of wake balancing. Moreover, this cost is small since the wake is concentrated in the center of the domain and only a few boxes are non-empty. In a problem with homogeneous distribution of particles along the entire domain, one may expect an increased balancing cost.



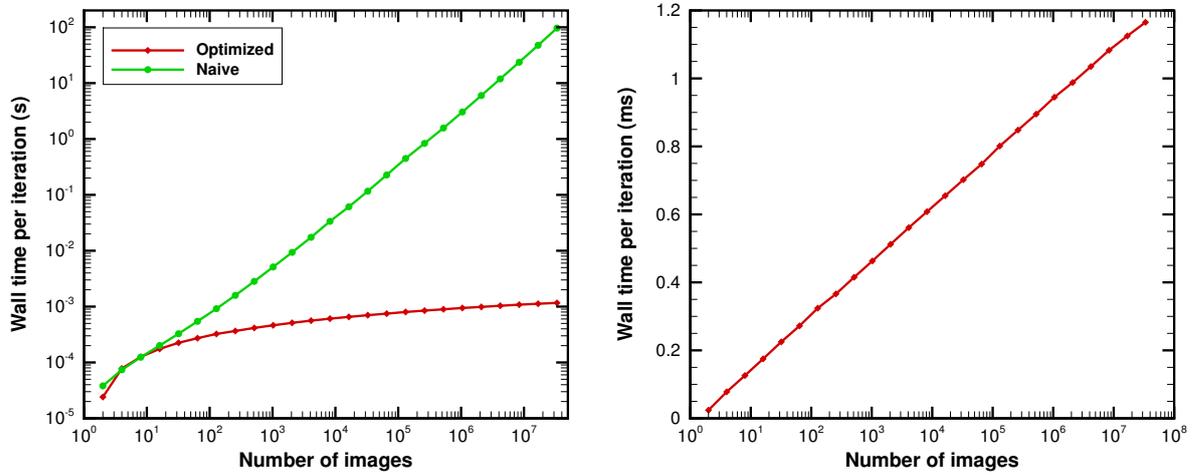
(a) Total computational time per iteration.

(b) Comparison of near-field cost (dashed lines) and balancing costs (solid lines).

Figure 5.23: Computational time for a single evaluation of all convection steps.

The computational time from M2L operations for all periodic domains with the central domain is also investigated. Comparing the optimized algorithm, Eq. 5.37, to the naive implementation, Eq. 5.38, it is possible to see a considerable time reduction in Fig. 5.24(a), shown in seconds. A detailed view on the cost of optimized scheme is shown in milliseconds in Fig. 5.24(b). This cost grows linearly with the logarithm of the number of images boxes,  $\mathcal{P}$ , as expected from Eq. 5.37. Furthermore, even for few particles in the simulation, the naive calculation of periodicity can be orders of magnitude higher than the other costs together and it may even be more expensive than the Biot-Savart direct summation.

More details on implementation of periodic FMM is shown in appendix B.



(a) Comparison of naive and optimized periodic algorithms. (b) Detailed view of the cost for the optimized scheme.

Figure 5.24: Computational time for a single evaluation of periodic images in FMM.

## 6 FAST COTANGENT FUNCTION SUMMATION

The idea in this chapter is to avoid the creation of the periodic level 0 domains and to work with a kernel that intrinsically represents the periodicity, using a single domain, to aim for higher accuracy. The cotangent function, which is the solution for a periodic array of vortices, can be evaluated using a series approach other than the simple replication and summation of vortex-particles shown in chapter 5. Since the region of convergence of any series plays an important role, some possibilities are discarded due to their limited convergence properties. With this in mind, two series expansions are studied in this chapter. The first one uses an exponential function to approximate the cotangent kernel and the second one uses a power series with coefficients given by Bernoulli numbers. For further details, one should see Abramowitz and Stegun (1964).

Also, in order to use fast summation methods to solve these alternative series, the original argument is replaced by two functions, being one for the observer and other for the source terms, Eq. 3.1. Hence, the computational cost goes from  $O(N^2)$  towards  $O(N)$ .

### 6.1 Exponential series expansion

The first possibility investigated is an exponential series expansion of the cotangent function given by

$$\cot(\pi z) = +i - 2i \sum_{n=0}^{\infty} \exp(2\pi i n z) , \quad (6.1)$$

where the argument  $z$  is given by

$$z = x + iy . \quad (6.2)$$

If one truncates the series with  $p$ -terms, it is possible to write an approximation as

$$\cot(\pi z) \approx +i - 2i \sum_{n=0}^p \left[ \exp(2\pi i n x) \exp(-2\pi n y) \right] . \quad (6.3)$$

Hence, two distinct exponential terms are present. The one dependent on the distance along the  $x$ -axis is imaginary and purely oscillatory, so it does not contribute to the convergence. On the other hand, the term based on the separation along the  $y$ -axis is real and decays if  $y > 0$  and, since  $|\exp(-2\pi y)|$  is smaller than unit, the series is convergent. For  $y < 0$ , there is divergence since the exponential grows.

However, as cotangent is an odd function, it is possible to achieve convergence for both  $|y| > 0$  based on

$$\cot(z) = -\cot(-z) . \tag{6.4}$$

This way, it is possible to write

$$\cot(\pi z) \approx +i - 2i \sum_{n=0}^p \left[ \exp(+2\pi inx) \exp(-2\pi ny) \right] , \text{ for } y > 0 , \tag{6.5}$$

and

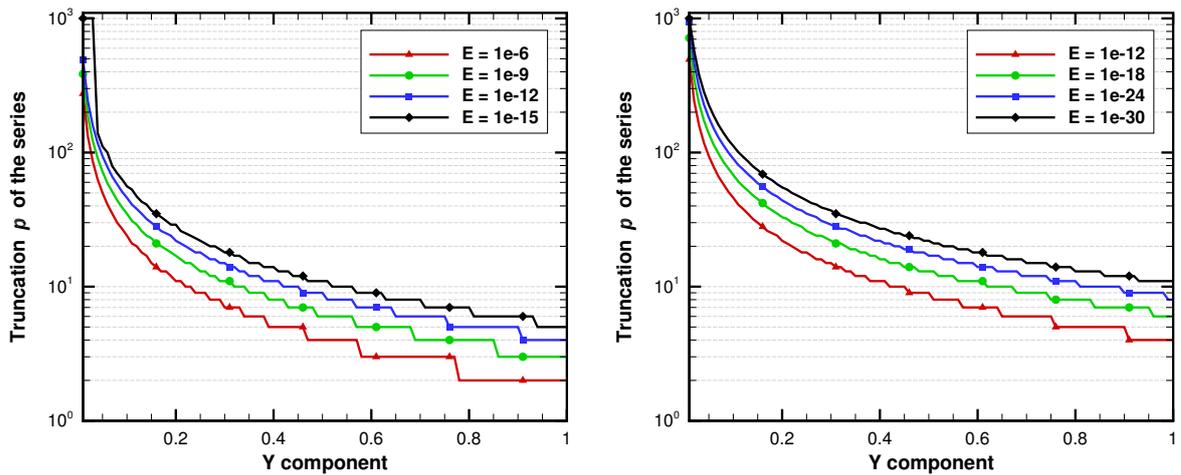
$$\cot(\pi z) \approx -i + 2i \sum_{n=0}^p \left[ \exp(-2\pi inx) \exp(+2\pi ny) \right] , \text{ for } y < 0 . \tag{6.6}$$

The series convergence is investigated to determine the number  $p$  of terms required to achieve an arbitrary precision  $E$ . To do so, one must solve

$$\left| \cot(\pi + i\pi y)_{x=1} - \left\{ i - 2i \sum_{n=0}^p \left[ \exp(2\pi in)_{x=1} \exp(-2\pi ny) \right] \right\} \right| \leq E , \tag{6.7}$$

which is valid for  $y > 0$ .

The results are indicated by the different curves in Fig. 6.1, ranging from  $E = 1 \times 10^{-6}$  to  $1 \times 10^{-30}$ . They are obtained for an argument where the imaginary part  $y$  ranges from 0 to 1, while the real part is fixed at  $x = 1$ . It is possible to see faster convergence for larger  $|y|$ , since fewer terms are required in the series expansion. The role of machine precision in presented in Fig. 6.1(a) and (b), for double and quadruple precision, respectively.



(a) Double precision.

(b) Quadruple precision.

Figure 6.1: Convergence properties of the first cotangent series.

## 6.2 Power series expansion

The second approximation for the cotangent function is given by a power series expansion for an argument  $|z| < 1$ , after a truncation with  $p$  terms, is given by

$$\cot(\pi z) \approx \sum_{n=0}^p \frac{(-1)^n 2^{2n} B_{2n}}{(2n)!} (\pi z)^{2n-1}, \quad (6.8)$$

where the Bernoulli numbers  $B_{2n}$  are defined as

$$B_{2n} = (-1)^{n+1} \frac{(2n)! 2}{(2\pi)^{2n}} \zeta(2n). \quad (6.9)$$

In the above equation, the Riemann zeta function  $\zeta$  is computed for  $n > 1$  as

$$\zeta(2n) = \sum_{i=1}^{\infty} i^{-2n}, \quad (6.10)$$

and, for  $n = 0$ , it is given by

$$\zeta(0) = -\frac{1}{2}. \quad (6.11)$$

Combining Eqs. 6.8 – 6.11, it is possible to obtain a simplified equation

$$\cot(\pi z) \approx \sum_{n=0}^p \frac{(-2) \zeta(2n)}{\pi^{2n}} (\pi z)^{2n-1}, \quad (6.12)$$

which leads to the cotangent power series equation

$$\cot(\pi z) = \frac{1}{\pi z} - \frac{1}{3}(\pi z) - \frac{1}{45}(\pi z)^3 - \frac{2}{945}(\pi z)^5 + \dots + O[(\pi z)^{2p+1}] \quad (6.13)$$

It is possible to simplify Eq. 6.12, leading to

$$\cot(\pi z) \approx \sum_{n=0}^p C_{2n-1} z^{2n-1}, \quad (6.14)$$

with coefficients  $C_{2n-1}$  that can be precomputed and stored for faster calculations

$$C_{2n-1} = \frac{(-2) \zeta(2n)}{\pi}. \quad (6.15)$$

The convergence of Eq. 6.14 is investigated for  $|z| < 1$ , where  $z = x + iy$ . In this computations, the imaginary part is null, *i.e.*,  $y = 0$ , but non-zero values for  $y$  can be used as well. Hence, only the influence of the separation of two particles along the  $x$ -axis

is investigated, such that  $z = x$ . The absolute error from the series expansion to the exact solution can be evaluated as

$$\left| \cot(\pi z) - \sum_{n=0}^p C_{2n-1} z^{2n-1} \right| = E . \quad (6.16)$$

Figure 6.2 shows that for small distances the cotangent behaves like  $1/z$ , and few terms are necessary for good convergence of the series. When the distance increases,  $z \rightarrow 1$ , the method lacks precision and the error grows, as presented in Fig. 6.3 for different number  $p$  of terms in the series. Finally, if the argument of the cotangent is larger than  $\pi$  ( $|z| > 1$ ), the present power series expansion is not applicable, *i.e.*, this formulation should not be used for vortex particles far from each other.

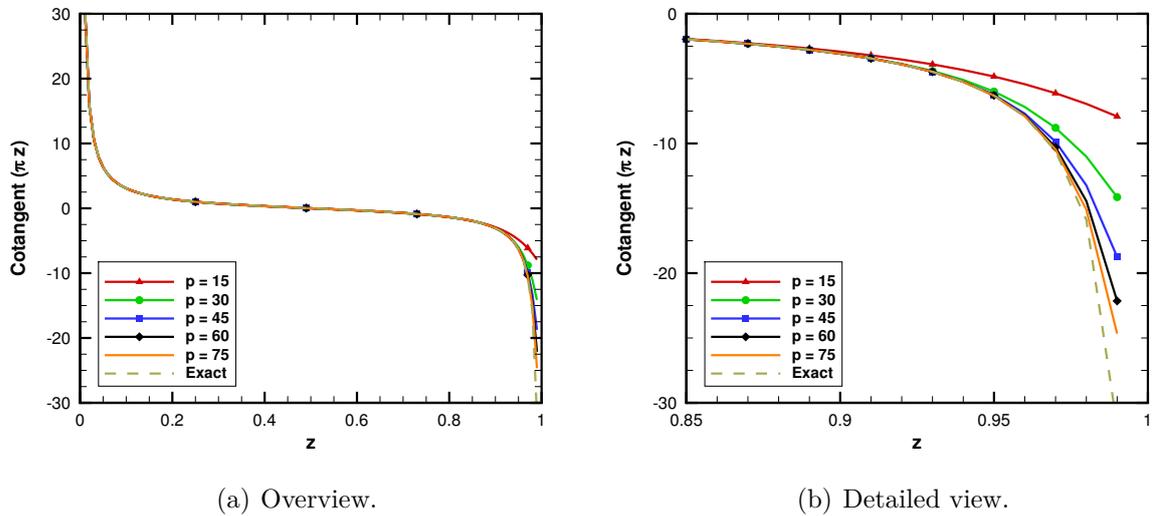


Figure 6.2: Absolute value of the cotangent power series expansion based on the argument  $z$  for several truncation terms  $p$  evaluated with double precision.

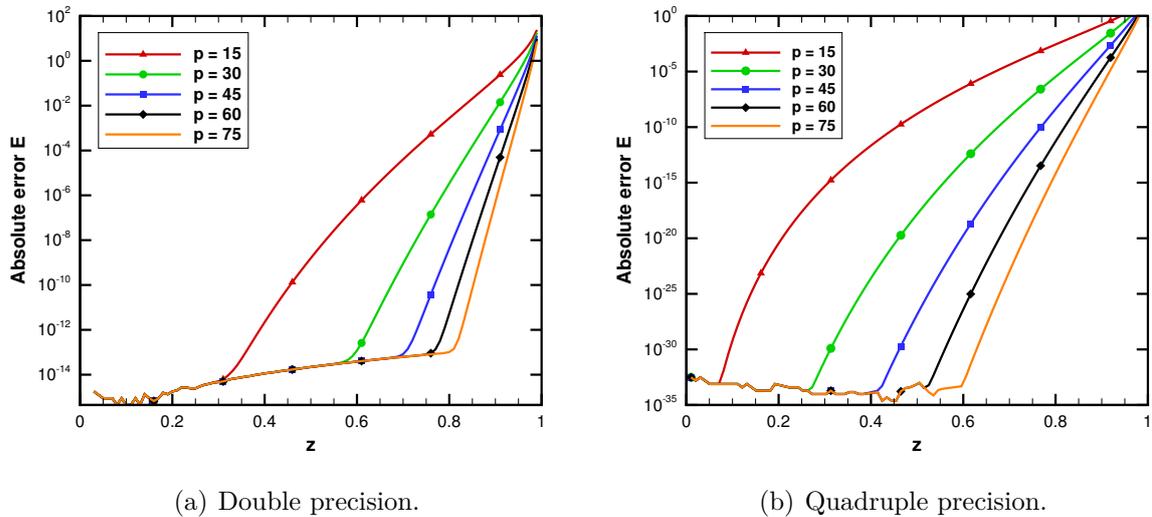


Figure 6.3: Error of the cotangent power series expansion based on the argument  $z$  for several truncation terms.

However, since the cotangent function is a periodic function with period  $\pi$ , it may be possible to rewrite it using integer numbers  $n$  ( $n \in \mathbb{Z}$ ) as

$$\cot(z) = \cot(z + n\pi) . \quad (6.17)$$

This way, when the distance of two particles along the real axis is large, it is possible to account their interaction not directly, but to use one of its image-vortex instead, leading to the same result. This idea is shown in Fig. 6.4 where, if  $|x| > 0.5$ , the interaction is performed not with the original vortex-particle, but with its periodic image. Hence, the convergence is faster everywhere inside an unitary domain.

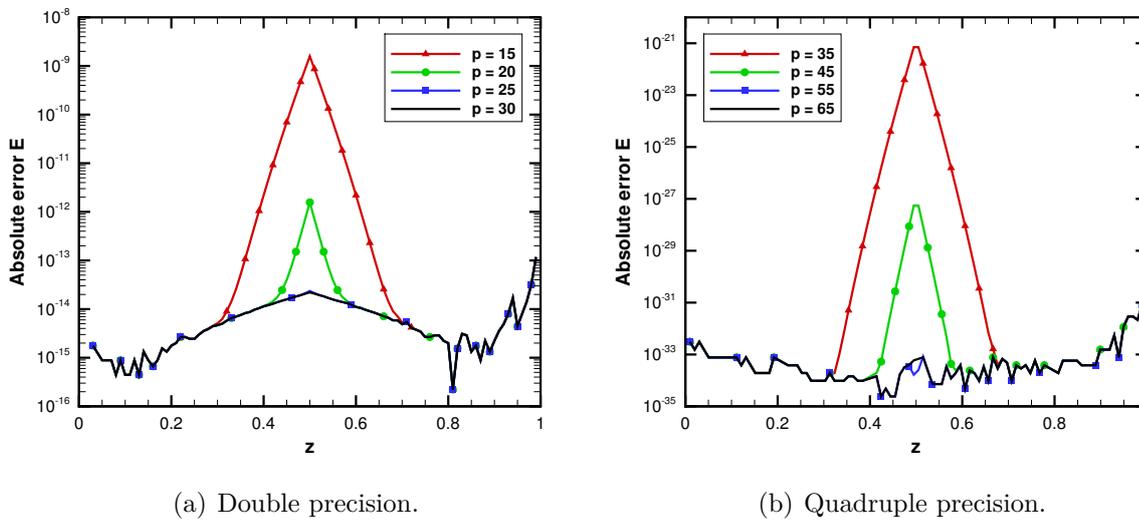


Figure 6.4: Convergence of the cotangent power series expansion.

### 6.3 Fast summation algorithm

In order to reduce the computational cost from the direct calculation of the Biot-Savart law from  $O(N^2)$  towards  $O(N)$ , a divide-and-conquer strategy will be used. Therefore, clusters of vortex particles should interact with each other instead of the direct particles. As presented in Figs. 6.1 – 6.4, particles close to each other are evaluated using the power series expansion. While the singular term  $1/z$  present in this series must be handled by the FMM, all the non-singular terms can be evaluated using a Newton binomial expansion. Hence, these two methodologies are coupled to solve the approximation through the power series expansion. The methodology of the FMM used here is the same presented in the previous chapters. For elements with large separation along the  $y$  axis, an exponential series expansion is employed.

The series expansion employed depends on the relative position among the boxes, so we implement an algorithm based on three methods to provide fast convergence everywhere.

### 6.3.1 Fast summation of the exponential series

Given a cluster  $b$  with  $N_b$  source-particles with intensity  $\Gamma_j$  placed at  $z_j$ , if one accounts for the particles and all their periodic images, the velocity field at a point  $z_i$  is given by

$$\sum_{j=1}^{N_b} \Gamma_j \cot(\pi z_{ij}) = \sum_{j=1}^{N_b} \Gamma_j \left\{ +i - 2i \sum_{n=0}^{\infty} \exp \left[ 2\pi i n (+z_i - z_j) \right] \right\}, \text{ for } y_i - y_j > 0, \quad (6.18)$$

or

$$\sum_{j=1}^{N_b} \Gamma_j \cot(\pi z_{ij}) = \sum_{j=1}^{N_b} \Gamma_j \left\{ -i + 2i \sum_{n=0}^{\infty} \exp \left[ 2\pi i n (-z_i + z_j) \right] \right\}, \text{ for } y_i - y_j < 0. \quad (6.19)$$

One must recall the requirement for convergence such that the imaginary part  $y$  is positive. If it is negative, the series given by Eq. 6.18 diverges. Hence, one must use the odd-function property of the cotangent, as explained in Eq. 6.4, to work with negative values of the imaginary part using Eq. 6.19. The deduction below is presented only for the case of positive imaginary part, but the modifications to work with negative values is straightforward.

In order to perform fast summations, following the idea of Eq. 3.1, it is necessary to split the argument in a product of two functions. This is straightforward for an exponential function

$$\exp(z_i - z_j) = \exp(+z_i) \exp(-z_j). \quad (6.20)$$

Furthermore, the *middleman* strategy of using the box centers as equivalent clusters can be employed. Defining  $z_{c_o}$  and  $z_{c_s}$  respectively as the center of observers and sources clusters, one can evaluate

$$\hat{z} = z_{c_s} - z_j, \quad (6.21)$$

$$\tilde{z} = z_{c_o} - z_{c_s}, \quad (6.22)$$

$$\bar{z} = z_i - z_{c_o}. \quad (6.23)$$

The equations above are, respectively, the distance from source cluster to the source-particles, the separation among clusters and the distance to observer-particles from the center of their clusters. From Eqs. 6.21 – 6.23, one recovers Eq. 6.20

$$\exp \left[ (z_{c_s} - z_j) + (z_{c_o} - z_{c_s}) + (z_i - z_{c_o}) \right] = \exp \left[ z_i - z_j \right]. \quad (6.24)$$

A similar technique of P2M, M2L, and L2P steps from the free-domain FMM can be employed. There is no need to use L2L steps since no multi-level operations are required.

From Eqs. 6.21 – 6.24, it is possible to rewrite Eq. 6.18 as

$$\sum_{j=1}^{N_b} \Gamma_j \cot(\pi z_{ij}) = \sum_{j=1}^{N_b} i\Gamma_j + \sum_{j=1}^{N_b} \left\{ -2i\Gamma_j \sum_{n=0}^p \left[ \exp(2\pi in\hat{z}) \exp(2\pi in\tilde{z}) \exp(2\pi in\bar{z}) \right] \right\}, \quad (6.25)$$

where, after manipulation of the order of the summations one has

$$\sum_{j=1}^{N_b} i\Gamma_j + \sum_{n=0}^p \left\{ -2i \exp(2\pi in\tilde{z}) \exp(2\pi in\bar{z}) \sum_{j=1}^{N_b} \left[ \Gamma_j \exp(2\pi in\hat{z}) \right] \right\}.$$

Therefore, a multipole-type expansion dependent only on the source-particles and defined for  $n = 0$  to  $p$  can be obtained using P2M operations as

$$\mathcal{M}_n^{(b)} = \sum_{j=1}^{N_b} \left[ \Gamma_j \exp(2\pi in\hat{z}) \right]. \quad (6.26)$$

Hence, one can write

$$\sum_{j=1}^{N_b} i\Gamma_j + \sum_{n=0}^p \left\{ -2i \exp(2\pi in\tilde{z}) \exp(2\pi in\bar{z}) \mathcal{M}_n^{(b)} \right\},$$

where it is possible to evaluate local representations for a box  $b'$  of far away clusters from the  $b$  boxes using M2L operations given by

$$\mathcal{L}_n^{(b')} = -2 \exp(2\pi in\tilde{z}) \mathcal{M}_n^{(b)}. \quad (6.27)$$

Finally, for  $n = 0$  to  $p$ , the use of L2P operations accounts for the induced velocity from all  $N_b$  source particles located in the far-field of a certain box  $b'$  at an observer placed at  $z_i$ . This is written as

$$\sum_{j=1}^{N_b} \Gamma_j \cot(\pi z_{ij}) = i \sum_{j=1}^{N_b} \Gamma_j + i \sum_{n=0}^p \left\{ \mathcal{L}_n^{(b')} \exp(2\pi in\bar{z}) \right\}. \quad (6.28)$$

Given the truncation of the series in  $p$  terms, the computational cost of P2M and L2P, Eqs. 6.26 and 6.28, respectively, is  $O(pN)$ . Hence, they are linearly dependent on the number of elements. Furthermore, the method has a fixed cost which depends only on the box-to-box operations given by Eq. 6.27. It is proportional to the desired precision as well as the number of non-empty boxes at a level  $\ell$ , given by  $O(4^\ell)$ . Hence, the exponential series summation has an overall cost of

$$O(2pN + 4^\ell p).$$

A quick test to validate this algorithm is performed with the computation of the absolute error  $E$ , based on the vertical distance for a far-field observer and series truncation using  $p$  terms. To do so, one thousand potential vortex particles are placed along the  $x$ -axis, while the observer is positioned along the  $y$ -axis, from  $y = -0.5$  to  $0.5$ . The solution obtained with the algorithm is then compared to the exact solution evaluated by the Biot-Savart law and the deviation is shown in Fig. 6.5. In the figure, each line represents a different number of truncation terms in the series, showing a convergence for quadruple machine precision.

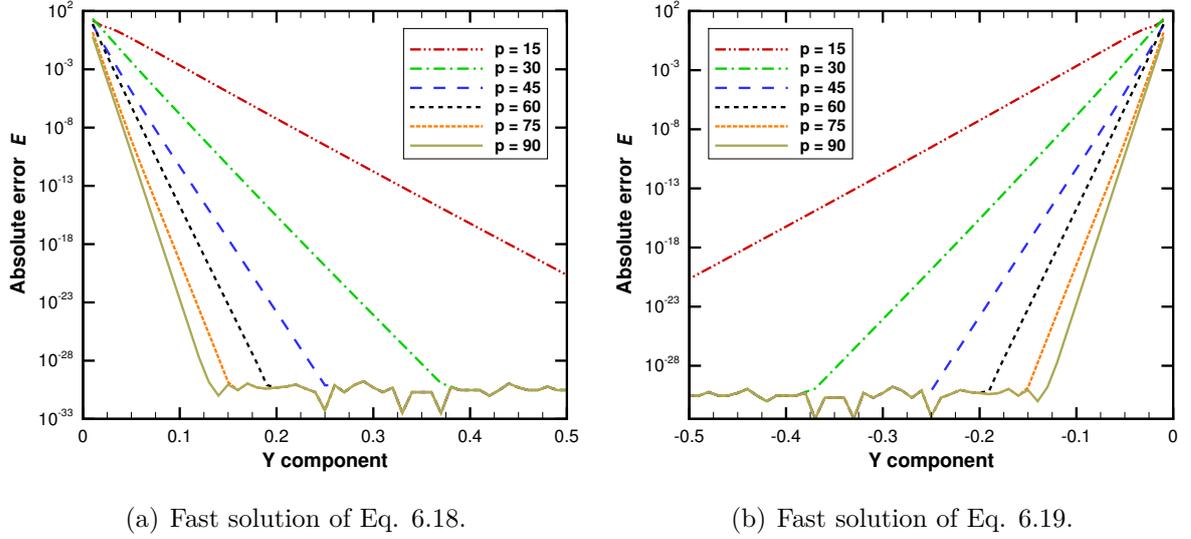


Figure 6.5: Error of the fast exponential series summation for different truncation terms.

### 6.3.2 Fast summation of the power series

The fast summation of the series given by Eq. 6.13 was first applied by Gumerov and Duraiswami (2004) to solve Fourier transforms. Here, it is applied to solve problems involving vortex dynamics using the discrete vortex method.

The local velocity field at  $z_i$  induced by the  $N_b$  periodic particles at  $z_j$  inside a box  $b$  is given by

$$w_i = i \sum_{j=1}^{N_b} \frac{\Gamma_j}{2\pi} \pi \cot(\pi z_{ij}) . \quad (6.29)$$

where  $z_{ij} = z_i - z_j$ . Two vortex particles can be far from each other in the central domain, but they can be close to the other's image. Hence, to have better convergence, the nearest distance among two vortex particles, using the property of periodicity of the cotangent

function, Eq. 6.17, is to compute  $z_{ij}$  as

$$z_{ij} = \begin{cases} z_{ij} - 1 & , \text{ if } |z_{ij} - 1| = \min(|z_{ij} - 1|, |z_{ij}|, |z_{ij} + 1|) \\ z_{ij} & , \text{ if } |z_{ij}| = \min(|z_{ij} - 1|, |z_{ij}|, |z_{ij} + 1|) \\ z_{ij} + 1 & , \text{ if } |z_{ij} + 1| = \min(|z_{ij} - 1|, |z_{ij}|, |z_{ij} + 1|) \end{cases} \quad (6.30)$$

Using Eq. 6.13 as an approximation for the velocity field leads to

$$w_i = \sum_{j=1}^{N_b} \left\{ \left( \frac{\Gamma_j}{2\pi} \right) \pi \left[ \frac{1}{\pi z_{ij}} - \frac{1}{3}(\pi z_{ij}) - \frac{1}{45}(\pi z_{ij})^3 - \frac{2}{945}(\pi z_{ij})^5 + \dots \right] \right\} , \quad (6.31)$$

where is possible to obtain two main terms in the induced velocity as

$$w_i = \sum_{j=1}^{N_b} \left\{ \frac{\Gamma_j}{2\pi z_{ij}} \right\} + \sum_{j=1}^{N_b} \left\{ \frac{\Gamma_j}{2\pi} \pi \left[ -\frac{1}{3}(\pi z_{ij}) - \frac{1}{45}(\pi z_{ij})^3 - \frac{2}{945}(\pi z_{ij})^5 + \dots \right] \right\} . \quad (6.32)$$

The first series involves the singular terms and they have exactly the same kernel used in the free-domain FMM,

$$\sum_{j=1}^{N_b} i \left\{ \frac{\Gamma_j}{2\pi z_{ij}} \right\} .$$

One should remind that is possible to avoid the singular values of the cotangent ( $z_{ij} = n\pi$ , for  $n \in \mathbb{Z}$ ) using the Lamb-Oseen vortex with a viscous core  $\sigma$ .

The second series contains the regular terms written as

$$\sum_{j=1}^{N_b} \left\{ \frac{\Gamma_j}{2\pi} \pi \left[ -\frac{1}{3}(\pi z_{ij}) - \frac{1}{45}(\pi z_{ij})^3 - \frac{2}{945}(\pi z_{ij})^5 + \dots \right] \right\} ,$$

with the coefficients shown in Eq. 6.15. After truncating the series using  $p$  terms, it is possible to use a simplified notation for this series given by

$$\sum_{j=1}^N \left\{ \frac{\Gamma_j}{2} \sum_{n=1}^p (C_{2n-1} z_{ij}^{2n-1}) \right\} .$$

The term  $z_{ij}^{2n-1}$ , for  $z_{ij} = z_i - z_j$ , can be solved exactly using a Newton binomial expansion

$$z_{ij}^{2n-1} = \sum_{m=0}^{2n-1} \binom{2n-1}{m} z_i^{2n-1-m} (-z_j)^m . \quad (6.33)$$

Considering a complex distance  $z_{ij}$  and swaping the summation order, it is possible

to use a Newton's binomial expansion and the series becomes

$$\frac{1}{2} \sum_{n=1}^p \left\{ C_{2n-1} \sum_{m=0}^{2n-1} \left[ \binom{2n-1}{m} z_i^{2n-1-m} \sum_{j=1}^{N_b} (\Gamma_j(-z_j)^m) \right] \right\} .$$

Since the inner summation depends only on the source-particle  $j$ , a multipole cluster  $\mathcal{B}_m^{(b)}$  can be defined for a box  $b$  as

$$\mathcal{B}_m^{(b)} = \sum_{j=1}^{N_b} (\Gamma_j(-z_j)^m) . \quad (6.34)$$

Furthermore,  $\mathcal{B}_m^{(b)}$  can be precomputed in order to avoid a convolution in the vortex particles. For  $m$  from 0 to  $2p-1$ , its computational cost is  $O(2pN)$ . From this new variable, it is possible to write the power series as

$$\frac{1}{2} \sum_{n=1}^p \left\{ C_{2n-1} \sum_{m=0}^{2n-1} \left[ \binom{2n-1}{m} z_i^{2n-1-m} \mathcal{B}_m^{(b)} \right] \right\} .$$

The cotangent series, Eq. 6.29, is written in its final form as

$$w_i = \sum_{j=1}^{N_b} \left\{ \frac{\Gamma_j}{2\pi} \frac{1}{z_{ij}} \right\} + \frac{1}{2} \sum_{n=1}^p \left\{ C_{2n-1} \sum_{m=0}^{2n-1} \left[ \binom{2n-1}{m} z_i^{2n-1-m} \mathcal{B}_m^{(b)} \right] \right\} , \quad (6.35)$$

where the free-domain FMM is employed to solve the singular term.

The most important time reduction in fast summation methods is due to the cluster-cluster operations, if the number of clusters is smaller than the number of particles. The function which governs the interaction among particles can be manipulated without loss of precision using the centroids of the clusters as *middleman* for the interactions. This is shown in Eqs. 6.20 – 6.24 for the exponential series expansion.

For the power-series, this strategy is not applicable since one does not have explicit functions of the centroid of the source and observer clusters (at least its solution is not known to the author). Hence, in this binomial series, the local expansions are performed via cluster-particle, where the number of particles is now relevant. In other words, at level  $\ell$ , there are  $O(4^\ell)$  non-empty boxes which interact with  $N$  particles. Moreover, due to the desired precision in the cotangent series evaluation, the inner and outer summation costs are, respectively,  $O(2p)$  and  $O(p)$ . Finally, adding the cost to create the clusters  $\mathcal{B}$ , the total cost of the binomial series expansion is

$$O(4^\ell 2p^2 N + 2pN) .$$

In order to validate this algorithm and to define parameters for the computations, a measurement of the error is performed. Also, an assessment of the number of discrete potential vortex particles in the computations is performed. A shear-layer is placed along  $y = 0$  in order to evaluate its self-induced velocity. The singular term of the cotangent series is solved using the FMM series expansion truncated with 40 and 75 terms for double and quadruple precision, respectively. In Fig. 6.6, the RMS deviation from the exact solution is shown. There, the dashed lines indicate computations in double precision (8 bytes), while solid lines represent results with quadruple precision (16 bytes). It is possible to see that the series diverges so the error increases when more terms are used in the series. It only converges to machine precision for the case of two particles.

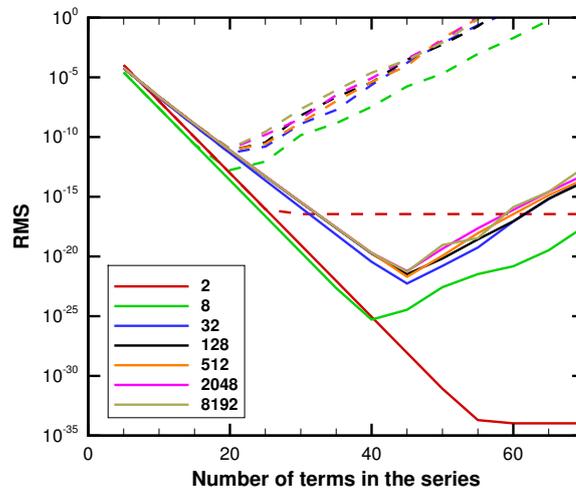


Figure 6.6: Error in velocity magnitude for a shear-layer, discretized by different numbers of vortices for double (dashed lines) and quadruple (solid) machine precision.

Although the series is convergent in Figs. 6.3 and 6.4, there, its calculation is performed using a straightforward evaluation of Eq. 6.13. Hence, each term in the power series goes to zero as the exponent becomes larger since there are no source errors. However, the fast summation algorithm given by Eq. 6.35 employs a Newton binomial expansion which may have truncation in several operations. For instance, the grouping of source-terms in  $\mathcal{B}_m$ , Eq. 6.34, as well as the evaluation of binomial coefficients, Eq. 3.23. This last term is the main source of error since it quickly becomes large enough for the computer to handle all its digits (the largest factorial without truncation for double and quadruple precision is, respectively, 21 and 37). Hence, from these points, the Newton binomial expansion may be incorrectly computed due to machine truncation. Consequently, the cotangent series has the largest errors in the algorithm and, in order to reduce these errors, the number of terms used in the series expansion is that which is closer to the inflexion point in Fig. 6.6.

### 6.3.3 Refinement of the initial domain

As explained, the separation among vortex particles plays an important role in the present fast algorithm. Based on Fig. 6.3 and 6.5, it is necessary that the minimum separation of the imaginary part to use the exponential series, as well as the argument  $z$  of the exponential series should obey

$$\begin{aligned} |y| &\gtrsim 0.125 && \text{to use exponential series or} \\ |z| &\lesssim 0.70 && \text{to use power series.} \end{aligned}$$

An efficient algorithm for fast summation of randomly scattered vortex particles should be based on clustering. Hence, the domain refinement idea from the FMM is employed here as previously discussed. One must not forget the convergence criterion for Eq. 6.8,  $|z| < 1$ , which imposes the largest domain to have unitary size, *i.e.*,  $S_0 = 1$ .

Using the ideas of divide-and-conquer, the domain is refined in smaller boxes which interact among each other. The separation of the boxes depends exclusively on the division of the domain. Using the same scheme of refinement from the free-domain FMM, in order to satisfy simultaneously the requirements of separation for both series discussed above, the refinement level used should be  $\ell = 3$ . In this refinement level, the size of a box is  $S_3 = 0.125S_0$ .

A methodology can be imposed to regulate the series expansion to be solved. The interactions among clusters with minimum vertical separation  $|y| = 0.125$  is performed via exponential series. In other words, the separation is, at least, one box from the source to the observer cluster, as presented in Fig. 6.7 for a generic observer box. In this figure, one can see an observer particle indicated in green inside a dark gray box. This particle interacts with a source particle indicated in cyan within a light gray box and their separation is given by  $|y| = 0.125$ . Hence, all boxes with  $y_i - y_j > 0$  must solve Eq. 6.18, while the ones where  $y_i - y_j < 0$  are solved by Eq. 6.19.

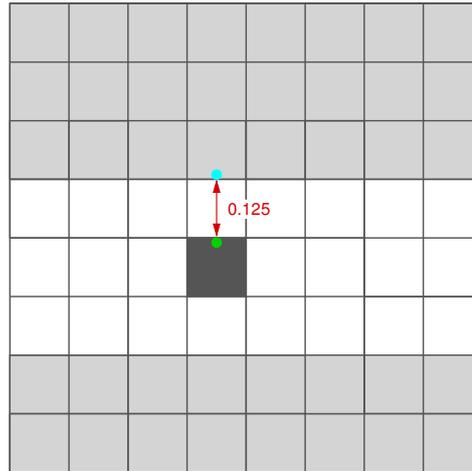


Figure 6.7: Light gray boxes in the exponential interaction list of that in dark gray.

In the region where the vertical separation  $|y|$  among two particles is smaller than 0.125, one must solve the power series expansion, Eq. 6.35. This is shown in Fig. 6.8, where a generic observer box indicated by dark gray interacts with all source boxes, colored by light gray. Furthermore, the maximum separation  $z$  among two particles is 0.673 and it guarantees good convergence based on Fig. 6.3.

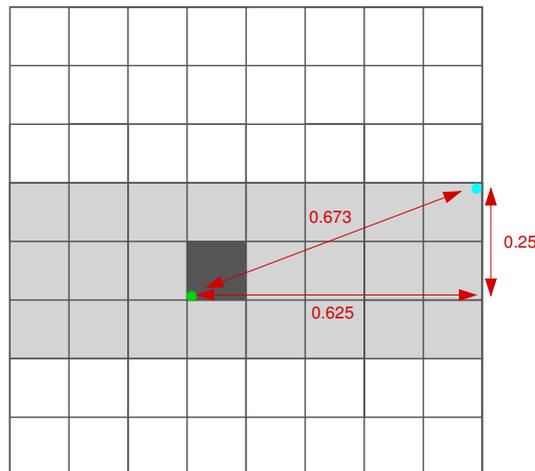


Figure 6.8: Light gray boxes in the power-series interaction list of that in dark gray.

If the distance among two particles is greater than the convergence range of Eq. 6.1,  $z < 1$ , this series expansion can not be applied. However, due to the periodic properties of the cotangent function, one can use the replica of a source-particle instead, so the argument  $z$  is within the convergence range, as presented in Eq. 6.30. In order to implement this logic, if an observer box is close to the border of the domain, it should not interact with the original source-cluster, but instead, it interacts with the replicated source-box within the nearest level 0 neighbor. One should remind that this logic is only valid for the calculation using the power series. The exponential series does not have any limitation with respect to the separation along the  $x$  axis and, therefore, for this case, all interactions are performed within the original domain.

In Fig 6.9(a) below, the light gray boxes are descendants of the left level 0 neighbor of the central domain, *a.k.a.* domain  $n = -1$ , and they interact with the dark gray box in Fig. 6.9(b). This artifice forces that the interactions are always performed for the nearest clusters to guarantee convergence.

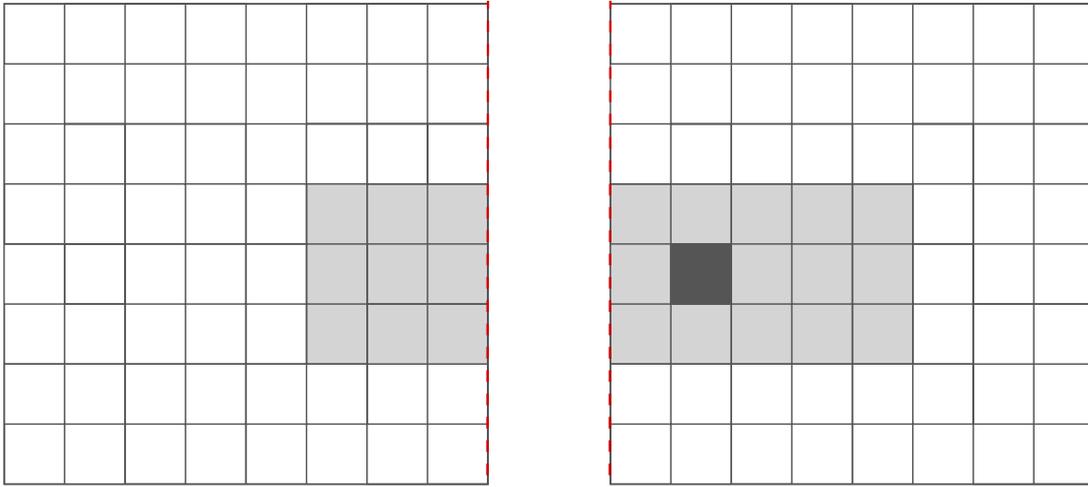
(a) Domain number  $n = -1$ .(b) Central domain  $n = 0$ .

Figure 6.9: Interaction list (light gray) of a box (dark gray).

One important remark is that level  $\ell = 3$  guarantees the fastest evaluation of both exponential series and Newton binomial expansion. The only drawback of such coarse level is the treatment of the singular term by the FMM, where direct evaluation of the cotangent function is still necessary in order to solve all terms of the power series. These direct evaluations are performed along near-field clusters which do not satisfy the separation criteria of well-separated boxes in the FMM.

In order to solve the FMM, one must solve both near and far-fields calculations. The fast summation of the far-field interactions is performed using box-to-box operations in a hierarchical pattern, where a multi-level approach and the interaction list of the FMM regulates the interactions among clusters. In the current implementation for the cotangent kernel, part of the domain is solved by the exponential series expansion, which does not require the evaluation of the  $1/z$  term. This singular term is present only in the power series, and, therefore, not all boxes interact via FMM inside the domain. Also, the interactions of both exponential and power series expansions are performed at level  $\ell = 3$ . For the two reasons presented, modifications in the interaction list are necessary to account all the 24 boxes, at level 3, marked to perform the interaction using power series expansion. The list of boxes for a generic observer box, indicated by dark gray in Fig. 6.8, is illustrated in light gray in the same figure. Also, to simplify the implementation, no operations at level  $\ell = 2$  are performed.

As pointed before in chapter 3, the near-field evaluation at level  $\ell = 3$  is expensive if

a large number of elements have to be accounted for. Therefore, further refinement levels are necessary to reduce this cost. Due to the criterion of separation to apply Eqs. 6.18 and 6.19, the 8 neighbors of a box at level 3 are always accounted for using power series, where the  $1/z$  term is present. Hence, these 9 boxes (the 8 neighbors and the box itself) can be recursively refined to level  $\ell = 4$  and forth, without any restriction. This is shown in Fig 6.10, where the boxes around the dark gray one have been further refined in a multi-level FMM. All the light gray boxes interact, either directly via M2L operations or indirectly with the combination of M2L and L2L operations from boxes at level 3, with the dark gray box. The white boxes are accounted for via exponential series and, hence, they are ignored in this step.

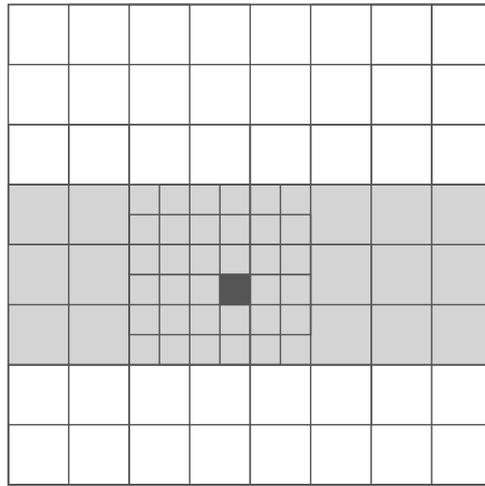


Figure 6.10: Refinement around the observer box for faster power-series solution.

It is worth noticing that the maximum level of the FMM algorithm to solve the near-field has no limitations in terms of separation criterion due to the series convergence. However the box can not be further refined if it conflicts with the particularities of each problem, *e.g.*, the viscous core of a Lamb vortex becomes larger than the box size.

Summarizing the fast algorithm: the fast interactions among particles are given either by Eq. 6.28 (and its counterpart that can be obtained from Eq. 6.19) if the particles are distant, or by the computation of Eq. 6.35. Although the exponential series is valid for positive arguments of  $\text{Im}(z)$ , one can benefit from the even-odd property of the cotangent function and work with a similar equation for negative arguments of  $\text{Im}(z)$ . Hence, one must compute first the boxes with positive separation and then evaluate the boxes with negative separation.

The power series is solved using two different methodologies: one is the FMM for the singular term  $1/z$  and the other one is the Newton binomial expansion which evaluates all the non-singular terms of the power series. The FMM still requires direct evaluations of the cotangent function while the Newton binomial expansion can be solved without restriction among neighbor boxes.. Also, due to the convergence criterion, boxes that are

far-away from each other may interact via one of their replicas, in order to guarantee convergence using the smallest argument.

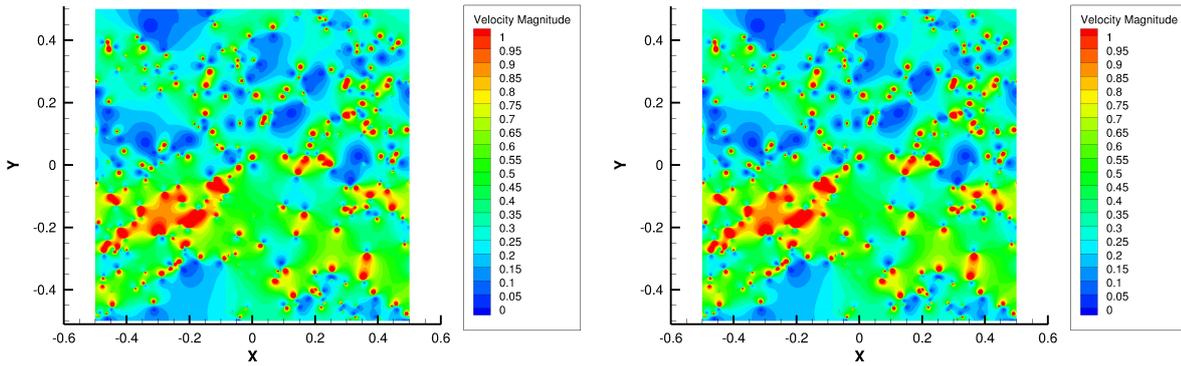
Also, the FMM can be accounted for using refinement levels  $\ell = 3$  or higher; the Newton binomial and the exponential series expansion are accounted for at  $\ell = 3$ . This level guarantees good convergence for all series and leads to the smallest computational time. Using this combination of free-domain FMM, Newton binomial expansion and fast summation of exponential series, three different formulations are employed simultaneously to solve the convolution among vortex particles with periodic boundary conditions for which the convolution kernel is the cotangent function.

## 6.4 Results

In this section, the fast algorithm is employed to solve a two dimensional velocity field with periodic boundary conditions using the DVM. Furthermore, an assessment of the error in velocity magnitude as well as the computational cost of the algorithm are shown. Results are presented firstly for a single time-step, where the self induced velocity field by a random cloud of vortices is evaluated. Finally, an assessment of computational time for the method is presented.

### 6.4.1 Evaluation of velocity field

To evaluate the velocity field, 256 vortex particles are placed randomly inside a square with unitary side, centered at the origin of the Cartesian coordinate system. The circulation of each vortex particle ranges from  $-0.1 \leq \Gamma \leq 0.1$ . In order to illustrate the problem, Figs. 6.11(a) and (b) show the velocity magnitude for the direct calculation solving the Biot-Savart law and the fast algorithm, respectively. The velocity is evaluated at a grid with 251 points in each direction. The deviation of the fast numerical algorithm is presented in Fig. 6.12, where the maximum error in velocity magnitude is about  $1 \times 10^{-12}$ .



(a) Biot-Savart solution. (b) Fast algorithm solution.

Figure 6.11: Magnitude of the velocity field.

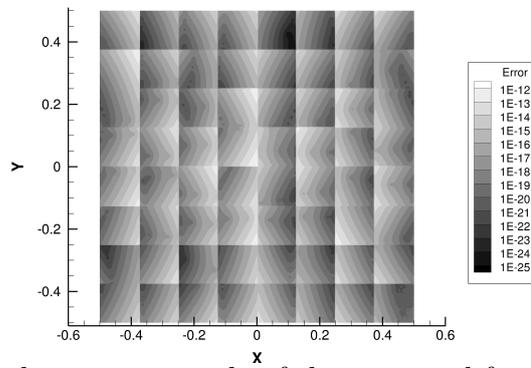
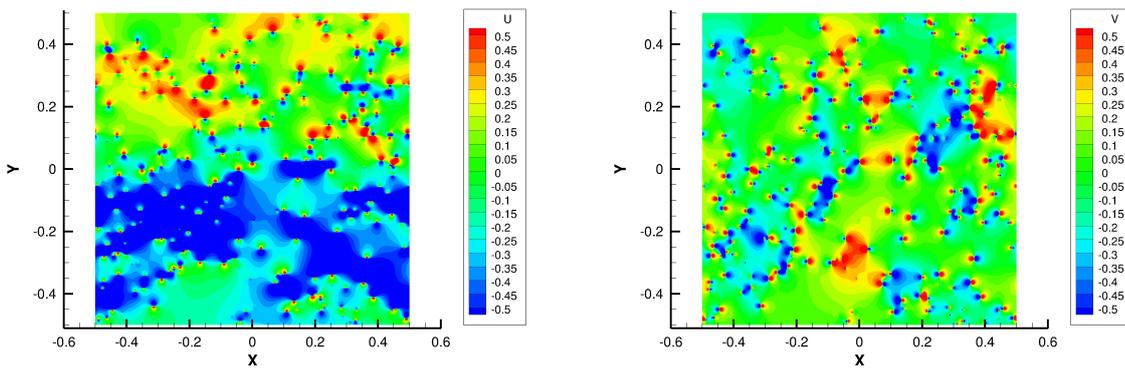


Figure 6.12: Error in velocity magnitude of the proposed fast algorithm compared to the Biot-Savart law computed using machine quadruple precision.

Also, it is possible to see both velocity components evaluated using only the fast method, shown in Figs. 6.13(a) and (b) for  $u$  and  $v$  components, respectively.



(a) U component. (b) V component.

Figure 6.13: Components of the velocity field evaluated by the fast summation method.

Finally, if one calculates the vorticity field, it is possible to locate the vortices since vorticity is only present inside the viscous core. This is shown in Fig. 6.14, where the blue dots are vortices in a clockwise sense and the red dots are those with counter-clockwise circulation.

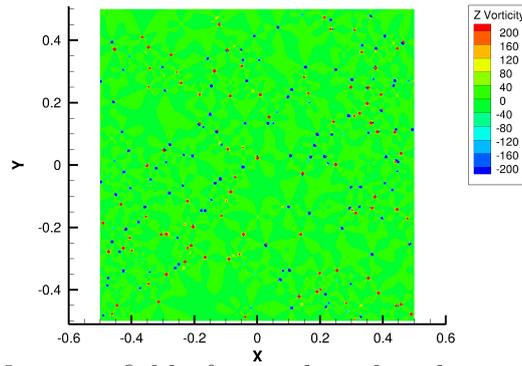


Figure 6.14: Vorticity field of a random distribution of vortex particles.

## 6.4.2 Error analysis

In order to measure the error based on the number of vortex elements, an investigation is performed from 2 up to 1 million vortex particles, randomly distributed inside the domain. The velocity magnitude is evaluated at each vortex location using the Biot-Savart law, the coupled fast method as well as the periodic scheme from section 5.1.3, which is a “brute-force” approach which replicates the FMM domains to simulate the periodic boundary conditions. Results from the brute-force approach are obtained using  $2^{30}$  FMM replicated boxes. The results of RMS deviation from a comparison of both fast methods to the Biot-Savart law are presented in Fig. 6.15. One can see that the cotangent approximation using coupled power and exponentials series is more accurate than the replicated FMM solution, called periodic FMM in the figure, and which has algebraic decay. Furthermore, this slow rate of decay implies that the usage of quadruple precision does not lead to better results. On the other hand, the coupled method benefits from increased machine precision, since the error drops six orders of magnitude. Finally, the RMS deviation of both methods grows proportional to  $O(N^{0.5})$ .

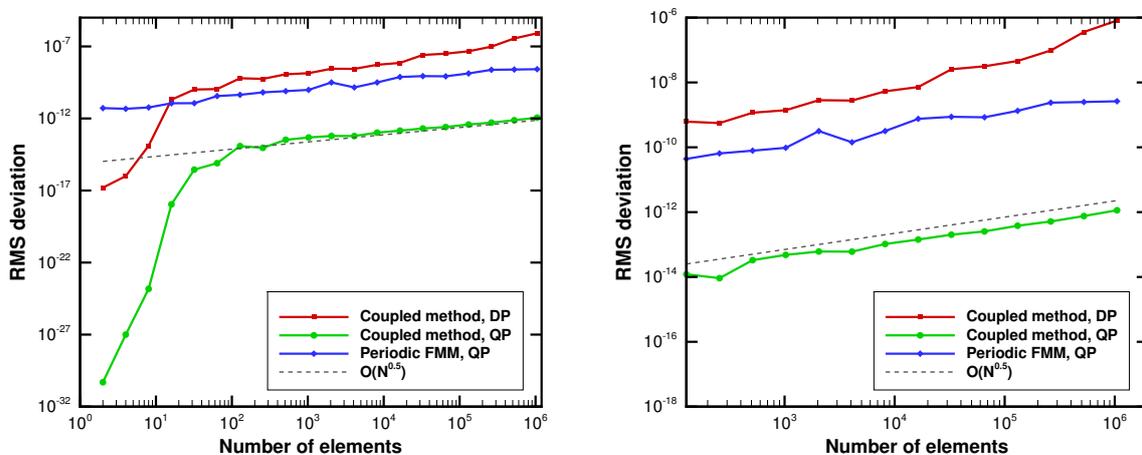


Figure 6.15: Error for a single evaluation of local velocity field for different methods.

### 6.4.3 Computational time

The overall computational cost of the algorithm results from the combination of the solutions of the free-domain FMM and the Newton binomial expansions or the exponential expansions. This section presents results of computational time for simulations performed in serial using a single thread in a 2.5 GHz Intel® Xeon™ E5-2670v2 CPU.

The results of overall computational cost are shown in Fig. 6.16(a) and (b) below for double and quadruple precision, respectively. The same behavior as the free-domain FMM is observed, where the increase in the number of elements require finer levels to reduce the quadratic dependence. This is shown by different curves in the legend, where the near-field is tested for different refinement levels in the FMM, which treats the singular term. One must not forget that the Newton binomial series and the exponential series are solved only at level 3. In the figure, one can see that for double precision (8 bytes) time savings of nearly 240 times are possible for 1 million elements, while in quadruple precision (16 bytes) the time savings for this number of elements is about 100 times. Furthermore, the fast algorithm using 16 bytes can be even faster than direct summation in double precision for more than 200000 vortex particles.

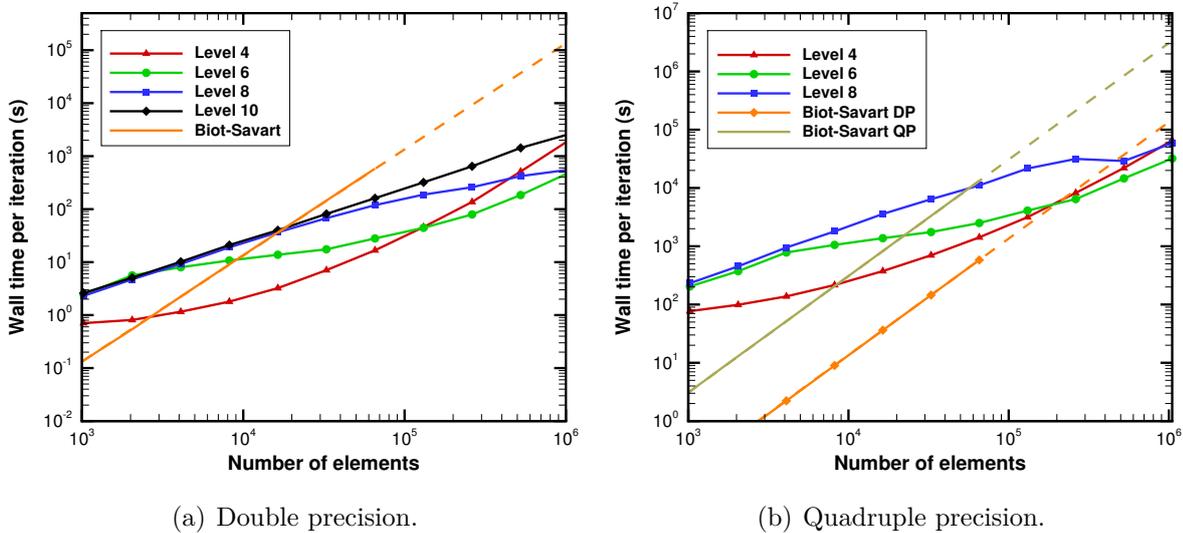


Figure 6.16: Overall computational cost of the combined method compared to the Biot-Savart solution.

The computational cost from the new algorithm is a combination of the costs from the free-domain FMM, the binomial series as well as the exponential series calculations. The first one is investigated in section 3.1.3, while the two new algorithm present a linear cost, dependent on the number of elements, and a fixed cost. Hence, the cost is given by

$$\mathcal{A}_2 N^2 + \mathcal{A}_1 N + \mathcal{A}_0 ,$$

where  $\mathcal{A}_0$  only depends on the number of non-empty boxes. For large refinement levels, there are several non-empty boxes for small number of elements, which delays the characterization of a fixed cost. This behavior is different than that presented in the temporal evolution of an aircraft wake due to the discretization of the problem. There, the number of non-empty boxes is constant and does not depend on the number of elements, so the parameter  $\mathcal{A}_0$  is a plateau, as shown in section 4.3.4.

On the other hand, as shown in Fig. 6.17, for a low number of particles in this random-cloud case, the coupled M2L and L2L operations have a delayed plateau and the cost  $\mathcal{A}_0$  is no longer independent of the number of elements. It is now based on the ratio of elements per number of box. If it is smaller than one, there are empty boxes indicating that fewer cluster operations are performed, so there is this initial slope on the curves. When there is at least one particle per box, without empty boxes, the domain becomes *saturated*. This way, the cost  $\mathcal{A}_0$  of box interactions is dumped for low number of particles. However, when the domain is saturated, the plateau appears and  $\mathcal{A}_0$  becomes constant.

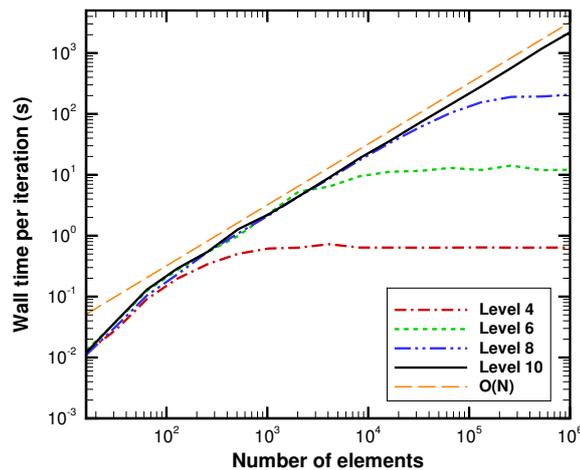
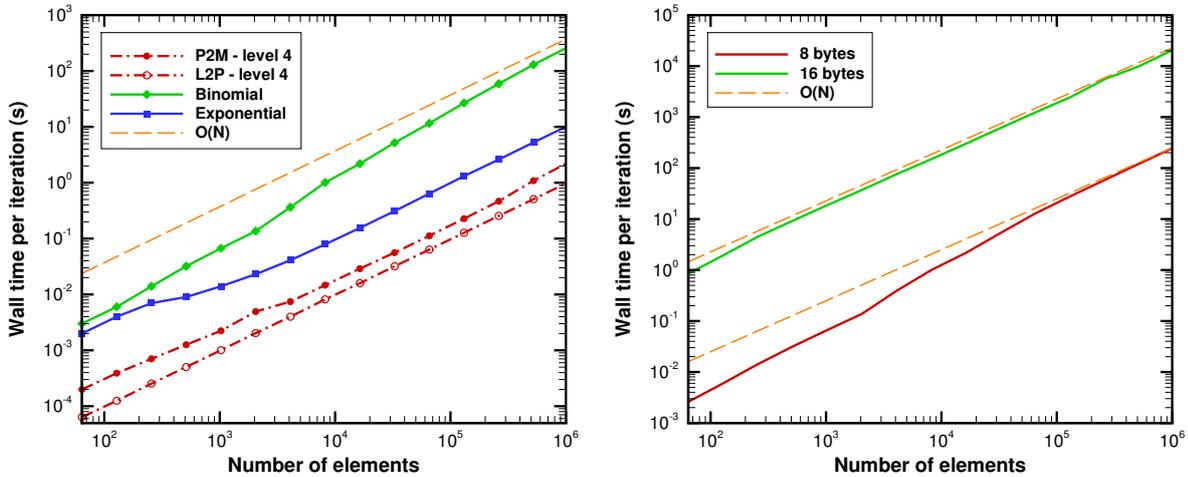


Figure 6.17: Computational time for box-to-box operations.

The additional operations to solve both exponential and power-series increase mainly the linear cost  $\mathcal{A}_1$ . These series are solved only at level  $\ell = 3$ , independently of the maximum refinement level  $L$  of the singular term solved by the free-domain FMM. Also, the P2M and L2P operations in the FMM do not depend on the maximum refinement level, and they are linearly proportional to the number of elements,  $O(pN)$ . For the exponential series, the additional cost is  $O(pN)$  as well. However, using the Newton binomial expansion at level  $\ell = 3$ , the additional cost  $O(4^3 \cdot 2p^2 \cdot N)$  is not only present, but also dominant for a small number of elements.

It is possible to see in Fig. 6.18(a) the most expensive operations dependent on the number of particles, *i.e.*, both P2M and M2L operations, as well the exponential and Newton binomial expansion for the power series. There, the computations are performed for double precision. In Fig. 6.18(b), one can see the cost of binomial expansion, for both

double and quadruple precision. There, this cost converges to  $O(N)$  when more elements are present in the domain.

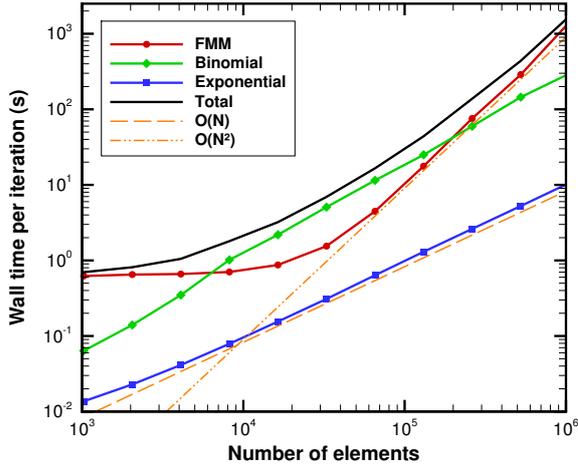


(a) Comparison of operations with linear cost. (b) Binomial expansion cost based on the precision.

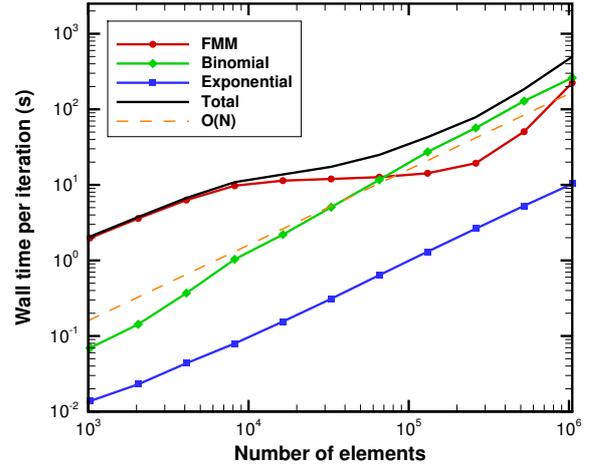
Figure 6.18: Computational cost of steps linearly dependent on the number of elements.

Finally, since both neither exponential nor power series have singular terms, elements close to each other interact exclusively via fast algorithms. This is not the case for FMM, which still depends on convolutions among elements close to each other. Hence, the quadratic dependency  $\mathcal{A}_2$  is the same as before and no further investigation for this cost is necessary.

In Fig. 6.19 it is possible to see a comparison of the cost of each individual algorithm used by the method. For a small number of elements, the far-field operations in the FMM are the most expensive. However, when the domain becomes *saturated*, the cost for the binomial series is the highest. Finally, for a large number of particles, the near-field cost in the FMM is dominant.



(a) Level 4



(b) Level 6

Figure 6.19: Computational cost of the three algorithms employed for fast summation of the cotangent function.

## 7 CONCLUSION

Lagrangian simulations of unsteady vortical flows are performed using the discrete vortex method, DVM, accelerated by the multi-level fast multipole method, FMM. Implementation details of the FMM-DVM algorithm are discussed for free domain and periodical problems including an analysis of the effects of temporal discretization and aspects of desingularization of vortex models in the DVM. An assessment of the parameters which control the errors of the FMM is performed showing the effects of refinement level and number of terms in the series expansion of the FMM.

The discretization of a continuous sheet using vortex particles leads to numerical errors, as pointed by Fink and Soh (1978). Along the roll-up region, the increase in the number of particles, either by a refined distribution towards the tip or by an overall increase in resolution, leads to more accurate solutions. The use of higher-order time marching schemes and smaller time steps also improves solutions. Therefore, we conclude that errors generated along this region are purely from convection mechanisms in the vorticity equation.

Another critical region in the shear layer is that portion where spurious instabilities appear. These phenomena are shown to be purely numerical and can be avoided when dissipation is added. It is demonstrated that the lack of artificial diffusion inherent of higher-order time marching schemes aggravates the instabilities. The increase in machine precision from double to quadruple precision, as well as the increase in the global number of elements in the cloud is shown to reduce truncation errors from discretization. Also, a diffusion model is shown to suppress the formation of instabilities, confirming the difficulty in working with non-dissipative methods.

Several other authors in literature have shown convergence studies for the Trefftz plane using the discrete vortex method. Using redistribution of the vortex elements eliminates the high frequency perturbations in the flow (Fink and Soh, 1978). The fast method proposed by Christiansen (1973) employs a mesh for the interpolation and interaction of vortices and it was used by Tryggvason (1989) to solve the vortex sheet roll-up in Trefftz plane. However, the method eliminates the instabilities through addition of non-controllable artificial dissipation.

Using the FMM, a completely meshless method originally developed for Coulomb interactions, it is possible to show solution convergence for the Trefftz plane problem for solutions that approximate the inviscid case, *i.e.*, when  $Re \rightarrow \infty$  or  $\sigma \rightarrow 0$ . This is achieved by increasing both the number of elements (reducing the discretization error)

and machine precision (reducing truncation error). However, it is still necessary to use small values of time step,  $\Delta t \rightarrow 0$ , to obtain a smooth shear layer.

The conclusions of this work agree with those already pointed out by Krasny (1987), showing the FMM capability to solve the problem. However, the desingularization parameters used by the author in his work are considerably higher than those investigated here, which added more diffusion to the problem. Here, the desingularization is kept as small as possible, so the time integration is performed for short periods, since sooner or later, spurious perturbations always arise.

The temporal evolution of a vortex-sheet roll-up was investigated for several parameters. Firstly, an inviscid shear layer is discretized by potential vortex. However, this situation is purely mathematical and in nature, this is not real since diffusion is present. A way to mimic a localized, controlled diffusion of vorticity is to use a desingularized vortex, *e.g.*, the Lamb-Oseen vortex model or the vortex blob by Krasny (1986a). However, in order to investigate the periodic FMM algorithm, the second model is not applicable together with this fast method. This way, only the Lamb-Oseen vortex was studied.

Smaller desingularization parameters are associated to less viscous effects. The computations of this quasi-inviscid cases leads to spurious perturbations of the shear layer, leading to wrong solutions. For instance, both Krasny (1986b) and Shelley (1992) applied Fourier filters to eliminate perturbations with arbitrary wavenumber. Although these authors obtain smooth solutions, in this work it was investigated other mechanisms to delay the perturbations. The usage of higher computer precision as well reduced sinusoidal displacement of the elements in the initial condition lead to a smooth solution. The increase in number of particles does not result in smoother solution.

After the solution from direct computations, an investigation of the usage of FMM in a periodic problem was performed. However, the replication of domains in the method is a crude approximation of the exact solution. A better approach is to use a balanced wake to avoid spurious precession of the wake, and it can be accelerated using FMM. This correction worked very well in this particular unidimensional problem with all vortex in the same orientation. However, its limitations on two or three dimensions, as well for random distributions of elements with any rotation sense are not known.

Finally, a fast summation of an alternative cotangent kernel is employed, where the solution of a power-series expansion as well as an exponential series avoids the replication of domains. Also, the balance use is not necessary anymore. This cotangent kernel results in reduced error compared to the crude replication of vortex particles.

## 7.1 Suggestions for future work

In this work, the convection is solved only by Runge-Kutta methods, which presents only the principal sigma root. Multiple-steps methods can also be employed, where the computational time to evaluate the velocity field does not depend on the method precision. Hence, this methods can reduce the time for simulations compared to RK schemes. However, these methods present not only the principal root but also spurious sigma roots which may compromise the solution. One can investigate the influence of this methods in the solution of non-linear model problems where both diffusion and vorticity generation are absent.

In terms of vorticity generation and presence of solid walls, the inverse matrix enables to work with multiple bodies with small increase in the computational cost. If direct methods are employed, the cost grows like  $O(n^3)$ , where  $n$  is the overall number of elements used to discretize the surfaces. Hence, problems with multiple geometries can be further studied.

The use of boundary conditions for periodic problems employed here is studied for one-direction only. Further tests can be performed in multi-dimensional problem to validate this technique in more complex situations.

Also in terms of periodic problems, the solution of the cotangent kernel is limited in a unitary domain. Further applications of this method must be investigated in situations where the domain employed requires larger dimensions. Also, the Newton binomial coefficient does not allow full convergence of the method and manipulations of this term must be proposed to avoid the loss of numerical precision when this term is employed.

## References

ABID, M. and VERGA, A. Stability of a vortex sheet roll-up. **Physics of Fluids**, vol. 14, 3829–3834, 2002.

ABRAMOWITZ, M. and STEGUN, I.A. **Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables**. National Bureau of Standards, U.S.A., 1964.

ANDERSON, C.R. An implementation of the fast multipole method without multipoles. **SIAM Journal on Scientific Computing**, vol. 13, n. 3, 923–947, 1992.

ANDERSON, C.R. and GREENGARD, C. On vortex methods. **SIAM Journal on Numerical Analysis**, vol. 22, n. 3, 413–440, 1985.

ANDRONOV, P.R.; GRIGORENKO, D.; S., G. and DYNNIKOVA, G. Numerical simulation of plate autorotation in a viscous fluid flow. **Fluid Dynamics**, vol. 42, 719–731, 2007.

APPEL, A.W. An efficient program for many-body simulation. **SIAM Journal on Scientific and Statistical Computing**, vol. 6, n. 1, 85–103, 1985.

AREF, H. Integrable, chaotic, and turbulent vortex motion in two-dimensional flows. **Annual Review of Fluid Mechanics**, vol. 15, 345–389, 1983.

ASHURST, W.T. Numerical simulation of turbulent mixing layers via vortex dynamics. In F. Durst; B.E. Launder; F.W. Schmidt and J.H. Whitelaw, editors, **Turbulent Shear Flows I - Selected Papers from the First Symposium on Turbulent Shear Flows**. 1977.

BAKER, G.R. and PHAM, L.D. A comparison of blob methods for vortex sheet roll up. **Journal of Fluid Mechanics**, vol. 547, 297–316, 2006.

BARBA, L.A., **Vortex Method for computing high Reynolds number flows: Increased accuracy with a fully mesh-less formulation**, PhD Thesis, California Institute of Technology, 2004.

BARBA, L.A. and COOPER, C.D. Panel-free boundary conditions for viscous vortex methods. **American Institute of Aeronautics and Astronautics Paper AIAA**, 2009.

BARBA, L.A.; LEONARD, A. and ALLEN, C.B. Numerical investigations on the accuracy of the vortex method with and without remeshing. **American Institute of Aeronautics and Astronautics Paper AIAA**, 2003.

BARNES, J. and HUT, P. A hierarchical  $O(N \log N)$  force-calculation algorithm. **Nature**, vol. 324, 446–449, 1986.

BATCHELOR, G.K. **An Introduction to Fluid Dynamics**. Cambridge Univ. Press, Cambridge, England, U.K., 2000.

BEALE, J.T. and MAJDA, A. High order accurate vortex methods with explicit velocity kernels. **Journal of Computational Physics**, vol. 58, 188–208, 1985.

BERNARD, P.S. A deterministic vortex sheet method for boundary layer flow. **Journal of Computational Physics**, vol. 117, 132–145, 1995.

BIMBATO, A.M.; ALCÂNTARA PEREIRA, L.A. and HIRATA, M.H. Simulation of viscous flow around a circular cylinder near a moving ground. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, vol. 31, n. 3, 243–252, 2009.

BIRKHOFF, G. Helmholtz and Taylor instability. **Proceedings of the Symposium on Applied Mathematics**, vol. 13, 1962.

BIRKHOFF, G. and FISHER, J. Do vortex sheets roll-up? **Rendiconti del Circolo Matematico di Palermo**, vol. 8, n. 1, 77–90, 1959.

BOX, G.E.P. and MULLER, M.E. A note on the generation of random normal deviates. **The Annals of Mathematical Statistics**, vol. 29, n. 2, 610–611, 1958.

CARRIER, J.; GREENGARD, L. and ROKHLIN, V. A fast adaptive multipole algorithm for particle simulations. **SIAM Journal on Scientific and Statistical Computing**, vol. 9, 669–686, 1988.

CHATELAIN, P.; CURIONI, A.; BERGDORF, M.; ROSSINELLI, D.; ANDREONI, W. and KOUMOUTSAKOS, P. Billion vortex particle direct numerical simulations of aircraft wakes. **Computer Methods in Applied Mechanics and Engineering**, vol. 197, 1296–1304, 2008.

CHEIN, R. and CHUNG, J.N. Discrete-vortex simulation of flow over inclined and normal plates. **Computers and Fluids**, vol. 16, 405–427, 1988.

CHENG, H.; GREENGARD, L. and ROKHLIN, V. A fast adaptive multipole algorithm in three dimensions. **Journal of Computational Physics**, vol. 155, 468–498, 1999.

CHORIN, A.J. Numerical study of slightly viscous flow. **Journal of Fluid Mechanics**, vol. 57, 785–796, 1973.

CHORIN, A.J. Vortex sheet approximation of boundary layers. **Journal of Computational Physics**, vol. 27, 428–442, 1978.

CHORIN, A.J. and BERNARD, P.S. Discretization of a vortex sheet, with an example of roll-up. **Journal of Computational Physics**, 1973.

CHRISTIANSEN, J.P. Numerical simulation of hydrodynamics by the method of point vortices. **Journal of Computational Physics**, vol. 13, 363–379, 1973.

CIPRA, B.A. The best of the 20th century: Editors name top 10 algorithms. **SIAM News**, vol. 33, 1–2, 2000.

COIFMAN, R.; ROKHLIN, V. and WANDZURA, S. The fast multipole method for the wave equation: A pedestrian prescription. **IEEE Antennas and Propagation Magazine**, vol. 35, 7–12, 1993.

DARVE, E. The fast multipole method: Numerical implementation. **Journal of Computational Physics**, vol. 160, 195–240, 2000.

DEHNEN, W. A hierarchical  $O(N)$  force calculation algorithm. **Journal of Computational Physics**, vol. 179, 27–42, 2002.

DEWEY, P.A.; QUINN, D.; BOSCHITSCH, B.M. and SMITS, A.J. Propulsive performance of unsteady tandem hydrofoils in a side-by-side configuration. **Physics of Fluids**, vol. 26, 88–109, 2014.

DRELA, M. **Flight Vehicle Aerodynamics**. The MIT Press, 2014.

DYNNIKOV, Y.; DYNNIKOVA, G. and S., G. Application of method vvd for a flow-structure interaction simulation of bodies with elastic connections. **International Conference on Vortex Flows and Vortex Models**, 2014.

ELDREDGE, J.D., **A Dilating Vortex Particle Method for Compressible Flow with Application to Aeroacoustics**, PhD Thesis, California Institute of Technology, 2002.

ELDREDGE, J.D. Efficient tools for the simulation of flapping wing flows. **American Institute of Aeronautics and Astronautics Paper AIAA**, 2005.

ELDREDGE, J.D.; COLONIUS, T. and LEONARD, A. A vortex particle method for compressible flows. **American Institute of Aeronautics and Astronautics Paper AIAA**, 2001.

FINK, P.T. and SOH, W.K. A new approach to roll-up calculations of vortex sheets. **Proceedings of the Royal Society of London**, vol. 362, 195–209, 1978.

GERRARD, J. Numerical computation of the magnitude and frequency of the lift on a circular cylinder. **Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences**, 1967.

GHONIEM, A.F. Grid-free simulation of diffusion using random walk methods. **Journal of Computational Physics**, vol. 61, 1–37, 1985.

GHONIEM, A.F.; HEIDARINEJAD, G. and KRISHNAN, A. Numerical simulation of a thermally stratified shear layer using the vortex element method. **Journal of Computational Physics**, vol. 79, 135–166, 1988.

GINEVSKY, A.S. and ZHELANNIKOV, A. **Vortex Wakes of Aircrafts**. Springer, 2009.

GOLIA, C. and VIVIANI, A. Airfoil flow analysis of a high lift system with the vortex blob method. **American Institute of Aeronautics and Astronautics Paper AIAA**, 2011.

GREENGARD, C. The core spreading vortex method approximates the wrong equation. **Journal of Computational Physics**, vol. 61, 345–348, 1985.

GREENGARD, L. and ROKHLIN, V. A fast algorithm for particle simulations. **Journal of Computational Physics**, vol. 73, 325–348, 1987.

GUMEROV, N.A. and DURAISWAMI, R. Fast multipole method based filtering of non-uniformly sampled data. Technical report, University of Maryland, 2004.

GUMEROV, N.A. and DURAISWAMI, R. **Fast Multipole Methods for the Helmholtz Equation in Three Dimensions**. Elsevier, 2005.

GUVERNYUK, S. and DYNNIKOVA, G. Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. **Fluid Dynamics**, vol. 42, 1–11, 2007.

HALD, O. and DEL PRETE, V.M. Convergence of vortex methods for euler's equations. **Mathematics of Computation**, vol. 32, 791–809, 1978.

HERRMANN, M. An eulerian level set/vortex sheet method for two-phase interface dynamics. **Journal of Computational Physics**, vol. 203, 539–571, 2005.

HESS, J.L. Calculation of potential flow about arbitrary three-dimensional lifting bodies. Technical Report MDC J5679/01, Douglas Aircraft Company, 1972.

HESS, J.L. and SMITH, A.M.O. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. Technical Report 40622, Douglas Aircraft Company, 1962.

HOLM, D.D.; NITSCHKE, M. and PUTKARADZE, V. Euler-alpha and vortex blob regularization of vortex filament and vortex sheet motion. **Journal of Fluid Mechanics**, vol. 555, 149–176, 2006.

HOU, T.Y.; LOWENGRUB, J. and KRASNY, R. Convergence of a point vortex method for vortex sheets. **Journal of Numerical Analysis**, vol. 28, 308–320, 1991.

KABADSHOW, I., **Periodic Boundary Conditions and the Error-Controlled Fast Multipole Method**, PhD Thesis, Bergische Universität Wuppertal, 2010.

KAMEMOTO, K. On contribution of advanced vortex element methods toward virtual reality of unsteady vortical flows in the new generation of CFD. **Journal of the Brazilian Society of Mechanical Science and Engineering**, vol. 26, 368–378, 2004.

KANTELIS, J.P. and WIDNALL, S.E. Calculations of axisymmetric vortex sheet roll-up using a panel and a filament model. Technical report, Massachusetts, 1986.

KATZ, J. and PLOTKIN, A. **Low Speed Aerodynamics: from Wing Theory to Panel Methods**. McGraw Hill, 1991.

KIDA, T. and NAKAJIMA, T. Core spreading vortex methods in two-dimensional viscous flows. **Computer Methods in Applied Mechanics and Engineering**, vol. 160, 1998.

KIYA, M. and ARIE, M. A contribution to an inviscid vortex-shedding model for an inclined flat plate in uniform flow. **Journal of Fluid Mechanics**, vol. 82, 223–240, 1977.

KOUMOUTSAKOS, P.; LEONARD, A. and PÉPIN, F. Boundary conditions for viscous vortex methods. **Journal of Computational Physics**, vol. 113, 52–61, 1994.

KOUMOUTSAKOS, P.D., **Direct Numerical Simulations of Unsteady Separated Flows Using Vortex Methods**, PhD Thesis, California Institute of Technology, 1993.

KRASNY, R. Desingularization of periodic vortex sheet roll-up. **Journal of Computational Physics**, vol. 65, 292–313, 1986a.

KRASNY, R. A study of singularity formation in a vortex sheet by the point-vortex approximation. **Journal of Fluid Mechanics**, vol. 167, 65–93, 1986b.

KRASNY, R. Computation of vortex sheet roll-up in the trefftz plane. **Journal of Fluid Mechanics**, vol. 184, 123–155, 1987.

KUDIN, K.N. and SCUSERIA, G.E. A fast multipole method for periodic systems with arbitrary unit cell geometries. **Chemical Physics Letters**, vol. 283, 61–68, 1998.

KUNDU, P.K.; COHEN, I.M. and DOWLING, D.R. **Fluid Mechanics**. Elsevier, 2012.

KURZAK, J. and PETTITT, B.M. Fast multipole methods for particle dynamics. **Molecular Simulation**, vol. 32, 775–790, 2006.

KUWAHARA, K. Numerical study of flow past an inclined flat plate by an inviscid model. **Journal of the Physical Society of Japan**, vol. 35, n. 5, 1545–1551, 1973.

LAMBERT, C.G.; DARDEN, T.A. and BOARD JR., J.A. A multipole-based algorithm for efficient calculation of forces and potentials in macroscopic periodic assemblies of particles. **Journal of Computational Physics**, vol. 126, 274–285, 1996.

LEONARD, A. Vortex methods for flow simulation. **Journal of Computational Physics**, vol. 37, 289–335, 1980.

LEONARD, A. Computing three-dimensional incompressible flows with vortex elements. **Annual Review of Fluid Mechanics**, vol. 17, 523–559, 1985.

LEWIS, R.I. **Vortex Element Method for Fluid Dynamic Analysis of Engineering Systems**. Cambridge Univ. Press, Cambridge, England, U.K., 1991.

LIENHARD, J.H. Synopsis of lift, drag, and vortex frequency data for rigid circular cylinders. Technical Report 300, Washington State University, 1966.

LOMAX, H.; PULLIAM, T.H. and ZINGG, D.W. **Fundamentals of Computational Fluid Dynamics**. Springer, 1999.

MALAKHOVA, T.; DYNNIKOVA, G. and S., G. Investigation of the heat transfer from oscillating cylinders by the vvhd method. **International Congress on Vortex Flow and Vortex Methods**, 2010.

MARTENSEN, E. The calculation of the pressure distribution on a cascade of thick airfoils by means of fredholm integral equations of the second kind. Technical Report 23, Max Planck Institute for Fluid Research and Aerodynamic Experimental Station, 1959.

MATSUOKA, C. Kelvin-Helmholtz instability and roll-up. **Scholarpedia**, vol. 9, n. 3, 2014. revision 143317.

MILINAZZO, F. and SAFFMAN, P. The calculation of large Reynolds number two-dimensional discrete vortices with random walk. **Journal of Computational Physics**, vol. 23, 380–392, 1977.

MOORE, D.W. The discrete vortex approximation of a finite vortex sheet. Technical Report AFOSR-1804-69, California Institute of Technology, 1971.

NAKANISHI, Y. and KAMEMOTO, K. Numerical simulation of flow around a sphere with vortex blobs. **Journal of Wind Engineering and Industrial Aerodynamics**, vol. 46 and 47, 363–369, 1993.

NISHIMURA, N. Fast multipole accelerated boundary integral equation methods. **Applied Mechanics Reviews**, vol. 55, 299–324, 2002.

NITSCHKE, M. and KRASNY, R. A numerical study of vortex ring formation at the edge of a circular tube. **Journal of Fluid Mechanics**, vol. 276, 139–161, 1994.

OGAMI, Y. Simulation of heat-fluid motion by the vortex method. **JSME International Journal**, vol. 44, n. 4, 513–519, 2001.

OGAMI, Y. Fast summation algorithms for the particle simulation of far-field potential and rapid decaying potential. **JSEM International Journal**, vol. 45, n. 1, 2002.

OGAMI, Y. and AKAMATSU, T. Viscous flow simulation using the discrete vortex model - the diffusion velocity method. **Computer and Fluids**, vol. 19, n. 3, 1991.

PLOUMHANS, P. and WINCKELMANS, G.S. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. **Journal of Computational Physics**, vol. 165, 354–406, 2000.

PLOUMHANS, P.; WINCKELMANS, G.S.; SALMON, J.K.; LEONARD, A. and WARREN, M.S. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at  $re = 300, 500, \text{ and } 1000$ . **Journal of Computational Physics**, vol. 178, 427–463, 2002.

PORTHOUSE, D.T.C., **Numerical Simulation of Aerofoil and Bluff Body Flows by Vortex Dynamics**, PhD Thesis, University of Newcastle upon Tyne, 1983.

RAMACHANDRAN, P.; RAMAKRISHNA, M. and RAJAN, S. Efficient random walks in the presence of complex two-dimensional geometries. **Computer and Mathematics with Applications**, vol. 53, 329–344, 2007.

RODIN, G.J. and OVERFELT, J.R. The point vortex approximation of a vortex sheet. **Proceedings of the Royal Society of London**, vol. 460, 2883–2902, 2004.

ROSENHEAD, L. The point vortex approximation of a vortex sheet. **Proceedings of the Royal Society of London**, vol. 134, 170–192, 1932.

ROSSI, L.F. Resurrecting core spreading vortex methods: A new scheme that is both deterministic and convergent. **SIAM J. Sci. Comput.**, vol. 17, 370–397, 1996.

SAFFMAN, P.G. **Vortex Dynamics**. Cambridge Univ. Press, 1992.

SARPKAYA, T. An inviscid model of two-dimensional vortex shedding for transient and asymptotically steady separated flow over an inclined plate. **Journal of Fluid Mechanics**, vol. 68, 109–128, 1975.

SARPKAYA, T. and SCHOAFFT, R.L. Inviscid model of two-dimensional vortex shedding by a circular cylinder. **AIAA Journal**, vol. 17, 1193–1200, 1979.

SETHIAN, J.A. On measuring the accuracy of the vortex method: On using a random method to model stable and unstable flow. In C. Anderson and C. Greengard, editors, **Lecture Notes in Mathematics**. Springer-Verlag, 1988.

SETHIAN, J.A. and GHONIEM, A.F. Validation study of vortex methods. **Journal of Computational Physics**, vol. 74, 283–317, 1988.

SHANKAR, S. and DOMMELEN, L.V. A new diffusion procedure for vortex methods. **Journal of Computational Physics**, vol. 127, 88–109, 1996.

SHELLEY, M.J. A study of singularity formation in vortex-sheet motion by a spectrally accurate vortex method. **Journal of Fluid Mechanics**, vol. 244, 493–526, 1992.

SHIELS, D., **Simulation of Controlled Bluff Body Flow with a Viscous Vortex Method**, PhD Thesis, California Institute of Technology, 1998.

SHINTANI, M. and AKAMATSU, T. Investigation of the two dimensional discrete vortex method with viscous diffusion model. **Computational Fluid Dynamics**, vol. 3, 237–254, 1994.

SHMAGUNOV, O.A. Modeling of jet flows of a viscous fluid by the discrete vortex method. **Journal of Applied Mechanics and Technical Physics**, vol. 53, 20–26, 2012.

SMITH, J.H.B. Vortex flows in aerodynamics. **Annual Review of Fluid Mechanics**, vol. 18, 221–242, 1986.

SPARLAT, P.R. Vortex methods for separated flows. Technical Report 100068, NASA Ames Research Center, 1988.

STOCK, M.J. Summary of vortex methods literature (a living document rife with opinion). 2007.

TAKAMI, H. Numerical experiment with discrete vortex approximation, with refence to the rolling up of a vortex sheet. Technical Report 202, Stanford University, 1964.

TRYGGVASON, G. Simulation of vortex sheet roll-up by vortex methods. **Journal of Computational Physics**, vol. 80, 1–16, 1989.

TRYGGVASON, G.; DHAM, W.J.A. and SBEIH, K. Fine structure of vortex sheet rollup by viscous and inviscid simulation. **Journal of Fluids Engineering**, vol. 80, 1–16, 1991.

TSUTSUI, T.; IGARASHI, T. and KAMEMOTO, K. Interactive flow around two circular cylinders of different diameters at close proximity. **Journal of Wind Engineering and Industrial Aerodynamics**, 1997.

WALTHER, J.H. and LARSEN, A. Two dimensional discrete vortex method for application to bluff body aerodynamics. **Journal of Wind Engineering and Industrial Aerodyamics**, vol. 67 and 68, 183–193, 1997.

WESTWATER, F.L. The rolling up of the surface of discontinuity behind an aerofoil of finite span. Technical Report 1692, British A. R. C. Reports and Memoranda, 1936.

WOLF, W.R., **Airfoil Aeroacoustics: LES and Acoustic Analogy**, PhD Thesis, Stanford University, 2011.

WONG, L.H., **A Numerical Model for Vortex Shedding from Sharp Wedges in Oscillatory Flow**, PhD Thesis, University of Newcastle Upon Tyne, 1985.

YOKOI, Y. and KAMEMOTO, K. Vortex shedding from an oscillating circular cylinder in a uniform flow. **Experimental Thermal and Fluid Science**, vol. 8, 121–127, 1994.

YOKOTA, R. and OBI, S. Comparing vortex methods and finite difference methods in a homogeneous shear flow. **International Journal for Numerical Methods in Fluids**, vol. 63, 828–846, 2010.

## APPENDIX A - FMM implementation

An overview of the current implementation of this FMM algorithm in fortran 90 is presented, highlighting some of its critical parts. The notation += indicates the summation on the variable to the left.

### A.1 Input

A few parameters must be defined for the Fast Multipole Method, mainly

```
naccur  ! is the number of terms in the Taylor series
nrefin  ! is the maximum refinement level
nimages ! is the number of "well-separated" image domains interacting
         ! with the central domain. Valid for periodic problems only.
```

Based on *nimages*, another variable called *ndomain*, which is only applicable for periodicity, indicates the central domain (*ndomain* = 0) and its left (*ndomain* = -1) and right neighbors (*ndomain* = +1). It is not null if there is periodicity, *i.e.*, if *nimages* is greater than 0.

### A.2 Pre-processing

Four subroutines are present: computation of boxes parameters, as well the list of children and parents; list of neighbors; list of interaction; and binomial coefficients. Also, it is performed a comparison of all particles to the level 0 FMM domain to check if any left the domain. If positive, only the first subroutine is called in order to enlarge the box. Furthermore, if the computational cost is expensive, the second and third subroutines only are called to refine the box. The fourth subroutine is computed only once, since it is only dependent on the precision.

In order to quickly obtain the neighbor and interaction lists, as explained in section 3.1.1, two variables indicate the parenthood relation of boxes: *nchild* and *nparent*.

```
allocate( nchild(0:nrefin-1, 1:4**nrefin, 1:4, -ndomain:ndomain))
allocate(nparent(1:nrefin, 1:4**nrefin))
```

The fourth index of *nchild* for a free-domain is only *ndomain* = 0. For periodic problems, it includes *ndomain* = ±1 to indicate the right/left neighbor of the central domain. This index is used to determine the descendant boxes in *ndomain* ±1 that are in the neighbor or interaction lists of descendant boxes of central domain.

There are two possibilities for allocation of the interaction list:

- to allocate naively like a matrix (0:L, 1:4<sup>L</sup>, 27) for the interaction list, with lots of positions zeros. For instance, a level 10 domain requires 1.16Gb of memory to allocate the interaction list for a free-domain FMM.
- to allocate using pointers, (i = 0:L)%(1:4<sup>i</sup>, 27) where empty positions are discarded. The same level 10 as above now requires 0.14Gb, only 12% of the previous cost.

In some cases, the memory cost of a variable is not significant and it may still be standardly allocated. However, in other cases, this cost may be prohibitive and more efficient ways are necessary to save memory. This way, for the interaction list, the second option is used and illustrated below with the pointer-variable *mult* to reduce the memory cost.

An auxiliary variable indicates how many boxes are in the interaction list of a box, in any level, to perform loops only in the useful boxes. As before, for periodic problems *ndomain* = ±1 indicates the descendants of the right/left neighbor of the central domain that are in the interaction list of a descendant box of the central domain.

```
do i = 1, nrefin
  allocate (mult(i)%inter(1:4**i, 1:27, -ndomain:ndomain))
enddo
allocate (nninter(1:nrefin, 1:4**nrefin, -ndomain:ndomain))
```

The neighbors list is allocated the same way as above, as well the auxiliary variable for the loops.

```
allocate (mult(0:nrefin))
do i = 0, nrefin
  allocate (mult(i)%neigh(1:4**i, 1:9, -ndomain:ndomain))
enddo
allocate (nneighb(0:nrefin, 1:4**nrefin, -ndomain:ndomain))
```

After allocating both neighbor and interaction lists, it is possible to create these lists based on *nchild* and *nparent* lists. In this current implementation, the boxes are numbered from the left to the right, beginning at the bottom. Hence, the first box is that in the left, at the bottom of the domain, while the last is the upper box at the right. With this in mind, the neighbor list can be quickly obtained since the 8 boxes around that of interest is easily obtained, plus the box itself. For boxes at the border of the level 0 domain, special attention must be taken. In periodic problems, these boxes are neighbors of others at domains  $\pm 1$ . In these cases, they are stored considering the third index.

```

do i = 0, nrefin
  do j1 = 1, 2**i                ! for all lines
    do j2 = 1, 2**i              ! for all columns
      j = (j1 - 1) * 2**i + j2   ! determine the box number "j"
      do k1 = j1 - 1, j1 + 1     ! line close to the box "j"
        do k2 = j2 - 1, j2 + 1   ! column close to the box "j"
          jj = (k1 - 1) * 2**i + k2 ! number of box close to "j"

          ! free-domain FMM
          if (k1 .le. 2**i .and. k1 .ge. 1 .and. &
              k2 .le. 2**i .and. k2 .ge. 1) then
            if (jj .ne. j) then
              !this is a neighbor box at ndomain = 0
            endif

            !periodic problem
            elseif (nimages .gt. 0) then
              if (k1 .le. 2**i .and. k1 .ge. 1 .and. k2 .gt. 2**i) then
                !this is a neighbor box at ndomain = +1
              endif
              if (k1 .le. 2**i .and. k1 .ge. 1 .and. k2 .lt. 1) then
                !this is a neighbor box at ndomain = -1
              endif
            endif
          endif

        enddo !k2
      enddo !k1
    enddo !j2
  enddo !j1
enddo !i

```

With the creation of the neighbor list, the second list for M2L interactions is straightforward for a box  $\mathbf{j}$  at level  $\mathbf{i}$ , such that  $\mathbf{i} \geq 1$ . For this box, an auxiliary variable,  $\mathbf{nv}$ , quickly indicates who are its neighbors, based on  $\mathbf{neigh}$ .

Then, a loop is performed for all the neighbors  $\mathbf{jjj}$  of  $\mathbf{j}$ 's parent, called  $\mathbf{jj}$ , which are at level  $\mathbf{i}-1$ . Then, a logical variable,  $\mathbf{flag}$ , check if any of all four children  $\mathbf{nnc}$  of the box  $\mathbf{jjj}$  are neighbors to  $\mathbf{j}$ . In case of false, *i.e.*, those that are not neighbors, are obligatorily boxes in the interaction list of  $\mathbf{j}$ , while the neighbors are neglected.

```

do l = -ndomain, ndomain
  do i = 1, nrefin
    do j = 1, 4**i

      ! neighbors of "j"
      nv = 0
      do m = 1, nneighb(i, j, l)
        nv(m) = mult(i)%neigh(j, m, l)
      enddo

      k = 0                                ! counter
      jj = nparent(i, j)                   ! parent of "j"
      do jj1 = 1, nneighb(i-1, jj, l)      ! neighbors of "jj"
        jjj = mult(i-1)%neigh(jj, jj1, l) ! neighbors of "jj"
        do nc = 1, 4                       ! four children of "jjj"

          ! check if any box "nnc" is also in "nv"
          nnc = nchild(i-1, jjj, nc, l)    ! four children of "jjj"
          flag = any(nnc .eq. nv(:))      ! .false. or .true.

          ! if "nnc" is not in "nv"
          if (flag .eqv. .false.) then    ! "nnc" is in the list
            k = k + 1                      ! counter
            mult(i)%inter(j, k, l) = nnc  ! "nnc" is the "k"-th box
          endif

        enddo !jjj1
      enddo !jj1
      nninter(i, j, l) = k                 ! number of boxes in the list
    enddo !j
  enddo !i
enddo !!

```

### A.3 Mapping

It is possible to determine which box at level  $i+1$  contain an element  $k$  comparing its position to the coordinates  $x$  and  $y$  of the center of a box  $j$  at level  $i$ , respectively  $xcbox$  and  $ycbox$ .

```

j = 1
do i = 0, nrefin - 1
  if (xp(k) .lt. xcbox(i, j, 0) .and. yp(k) .lt. ycbox(i, j, 0)) then
    box(k, i+1) = nchild(i, j, 1, 0)
  endif
  if (xp(k) .ge. xcbox(i, j, 0) .and. yp(k) .lt. ycbox(i, j, 0)) then
    box(k, i+1) = nchild(i, j, 2, 0)
  endif
  if (xp(k) .lt. xcbox(i, j, 0) .and. yp(k) .ge. ycbox(i, j, 0)) then
    box(k, i+1) = nchild(i, j, 3, 0)
  endif
  if (xp(k) .ge. xcbox(i, j, 0) .and. yp(k) .ge. ycbox(i, j, 0)) then
    box(k, i+1) = nchild(i, j, 4, 0)
  endif
enddo

```

The variable  $box$  stores the box in which the particles  $k$  is inside. This allows a quick way to relate boxes and particles, reducing computational cost when creating the multipoles or when translating the far-field influence among the particles of a box.

Furthermore, based on  $box$ , it is possible to know how many boxes are not empty in all refinement levels. The loops at level  $i$  in FMM steps will be performed from  $i$  to  $nbox(i)$ , and not from 1 to  $4^i$ . Then, the variable  $nbox\_list(i, j)$  relates the index  $j$  of the loop to the number of each not empty box. Also, the variables are allocated based on  $nbox(i)$  in order to reduce memory costs and allow higher refinements levels. This way, it is possible to reduce both  $\beta$  and  $\gamma$  coefficients related to the computational cost of the FMM, as presented in section 3.1.3.

## A.4 Upward pass

The pointer-variable *mult* is also used in order to reduce the memory cost for  $\mathbf{a}_k$ . The fastest way is to allocate a three-dimension variable for  $\mathbf{a}_k$  to store the multipole in all levels and do a  $O(N)$  operation only in the finer level. If one eliminates the third index, this variable has to be calculated with a cost  $O(N)$  for all levels. The first implementation tends to computational time  $O(N)$  while the second tends to  $O(N \log(N))$ .

```
do i = 0, nrefin
  allocate (mult(i)%ak(nbox(i), 0: naccur))
enddo
```

At the finer level, the clustering operations are:

```
do j = 1, nbox(nrefin)           ! loop in the not-empty boxes
  mult(nrefin)%ak(j, 0)         ! calculate ak for "k" = 0
  do k = 1, naccur
    nullify akaux               ! nullify "ak" for summation
    do i = 1, np(j)             ! for all particles inside a box
      akaux += P2M              ! inner summation
    enddo
    mult(nrefin)%ak(j, k) = akaux ! ak for "k" > 0
  enddo
enddo
```

The translation and clustering at coarser levels:

```
do i = nrefin - 1, 0, -1 ! from the finer levels to coarser
  do j = 1, nbox(i)      ! loop in the not-empty boxes
    do k = 1, 4          ! cluster the influence of the 4 child
      do n = 1, naccur ! for all required terms
        nullify akaux
        do m = 1, n
          akaux += M2M for mult(i+1)%ak(j, n)
        enddo
        mult(i)%ak(j, n) += auxak - M2M for mult(i+1)%a(k, 0)
      enddo
    enddo
  enddo
enddo
```

## A.5 Downward pass

As explained in Eq. 3.31, first the L2L operations are done from level  $\ell - 1$  to  $\ell$  followed by the M2L at level  $\ell$ . Contrarily to the upward pass, it is possible to allocate a two-dimensional variable for M2L without increasing computational time, and it is only necessary to create an auxiliary variable:

- At level 0, it is only necessary to calculate M2L and store in **bk**. After that, **bk** is transferred to the auxiliary **expd** and then nullified.
- At level 1, the L2L transfer values from **expd**, *i.e.* the M2L operations at level 0, to **bk**. The M2L operations, now at level 1, adds also to **bk** in order to accumulate the influence of level 0 and level 1. After all computations, **bk** is transferred to **expd** and nullified.
- At level 2 or greater, repeat the process above, but do not nullify **bk** in the finer level.

This way, at level  $\ell$ , **expd** has the values of far-field for all coarser levels (from 0 to  $\ell - 1$ ) and **bk** stores the far-field up to current level (from 0 to  $\ell$ ). These transfers are done to avoid conflict with the boxes in different levels with the same number.

```
allocate( bk(1:nbox(nrefin),0:naccur))
allocate( expd(1:nbox(nrefin),0:naccur))
```

For periodic problems, there is a M2L operation before the downward pass inside the central domain. The *well-separated* boxes at level 0, *i.e.*, domains  $\pm 2 \rightarrow \infty$  interacts with the central domain.

```
!interaction of periodic domains, except -1, 0 and +1
do k = -nimages, nimages      ! naive implementation
  do m = 1, naccur
    do n = 1, naccur
      bkaux += M2L at level 0 using mult(0)%ak(1,n)
    enddo
    bk(m) += bkaux - M2L using mult(0)%ak(1,0)
  enddo
enddo

expd(:, :) = bk(:, :)      !transfer "bk" to dummy "expd"
nullify bk
```

After the M2L at level 0, this influence is translated to the 4 child of central domain at level 1 from dummy variable *expd* using L2L steps. If there is not periodicity,  $\mathbf{a}_k$  is initially zero and does not add useful information.

```

do i = 1, nrefin
  do j = 1, nbox(i)          ! not empty boxes at level (i)
    do k1 = 0, naccur
      do k2 = k1, naccur
        bk = L2L from expd  ! L2L steps from dummy variable "expd"
      enddo !k2
    enddo !k1
  do l = -ndomain, ndomain  ! interaction of domains -1, 0 and +1

    ! perform the loop in all boxes in the interaction list of "j"
    do k = 1, nninter(i, j, l)
      do m = 1, naccur
        do n = 1, naccur
          bkaux += M2L(k)  ! M2L operations from "k" to "j" at "i"
        enddo !n

        ! sums the M2L and L2L, plus further operations
        ! eq 2.12 from Greengard & Rokhlin 1987
        bk(j,m) += operations with "bkaux" and "ak(k,0)"
      enddo !m
    enddo !k
  enddo !!
enddo !j

! transfer "bk" to "expd"
expd(:, :) = bk(:, :)
if (i .ne. nrefin) then ! "bk" must be null for the next level
  bk(:, :) = 0
endif
enddo !i

```

## A.6 Local-to-particle

With all far-field calculations, two loops are performed to transfer the influence from far-field to all particles, with cost  $O(Np)$ , for  $p$  being the size of the series. The distance  $z$  is from the observer particle to the center of the box,  $xcbox$  and  $ycbox$ .

```
do i = 1, npart
  j = box(i)
  z = zp(i) - cplx(xcbox(j), ycbox(j))
  do k = 1, naccur
    wff(i) = wff(i) + bk(j, k)*k*z**(k-1)
  enddo
enddo
```

## A.7 Particle-to-particle

After all far-field computations are performed, one should evaluate the near-field influence at a particle  $m$ . This is done among all the 8 neighbors as well inside the box itself. One must recall that in the case of a periodic problem, the neighbor boxes could be in the domains  $\pm 1$ . Using the list of neighbors,  $neigh$ , which indicates the  $j$ -th neighbor of a box. An auxiliary variable  $nneigh$  counts how many neighbors a box has. This direct evaluation is performed for all  $np$  particles inside a box  $k$ . This evaluation computes directly Biot-Savart law based on the position  $zp$  of the particles, as well on the circulation  $cp$  of the source. Also, one should use a smoothing function to overcome the singularity.

```
do l = -1, 1
  do j = 1, nneighb ! neighbors boxes
    k = neigh(j)
    do i = 1, np(k)
      wnf(m) = wnf(m) + cp(i)/(zp(m) - zp(i))*smoothing_function
    enddo
  enddo
enddo
k = box(m) ! inside box itself
do i = 1, np(k)
  wnf(m) = wnf(m) + cp(i)/(zp(m) - zp(i))*smoothing_function
enddo
```

## APPENDIX B - Periodic FMM

Details on the optimized implementation of FMM with periodic boundary conditions in one dimension is described in this appendix. Firstly, there are no further modifications in the preprocessing described in appendix A, only in box-to-box operations.

### B.1 Multipole-to-multipole

The periodic images well-separated from the central domain are clustered in a manner to reduce the computational time. The number of boxes to be further clustered are arbitrary, and, in this work, it is performed in order to create a cluster twice as that in the previous level. In other words, the clusters always double the size when the level increase. This eliminates variables and simplify implementation, since only two clusters, equally spaced from the center of the new cluster, are used in the computations.

```

do i = 0,n
  ak_0(i,0) = (2.0**i)*mult(0)%ak(1,0)  !see Eq. 5.35
enddo

do i = 1,n
  ! define two boxes at (i-1) to be clustered at (i)
  auxx = -0.5*2**(i-1)          ! see Eqs. 5.28, 5.33, 5.34
  z1 = cmplx(+auxx,0.0)        ! see Eq. 5.33
  z2 = cmplx(-auxx,0.0)        ! see Eq. 5.34
  do k = 1,naccur
    nullify akaux

    ! summation for the first term in Eq. 5.32
    do m = 1,k
      binom = binom_coeff(k-1,m-1)
      akaux += ak_0(i-1,m)*binom*(z1**(k-m) +z2**(k-m))
    enddo

    ! subtract the second term in Eq. 5.32
    ak_0(i,k) = akaux - (ak_0(i-1,0)/k)*(z1**k + z2**k)
  enddo
enddo

```

## B.2 Multipole-to-local

After the multipole-to-multipole operations for clustering periodic domains, one must evaluate the multipole-to-local operations from all these new clusters to the central domain using Eq. 5.34. However, manipulating this equation, one can see that, for odd indices, the multipole-to-local is anti-symmetric and the influence from right and left clusters, at the same level, cancel each other. Hence, it is possible to halve the computational time and only solve Eq. 5.34 for even indices.

```

do i = 0,n
  auxx = 1.5*2.0**(i) + 0.5      ! see Eq. 5.37 and 5.38
  auxy = 0.0                    ! center at y = 0.0
  auxz = cmplx(auxx,auxy)      ! see Eq. 5.37 and 5.38

  !only even terms are required
do m = 2,naccur,2
  nullify bkaux

  ! inner loop
do k = 1,naccur
  binom = binom_coeff(m+k-1,k-1)
  bkaux += ak_0(i,k)*binom*(-1.0)**k/(auxz**k)
enddo

  ! adds simultaneously both right and left periodic clusters
  ! avoids summation over 's' in Eq. 5.34.
  bk(1,kk) = 2.0*(bkaux - ak_0(i,0)/m)/auxz**m
enddo
enddo

```

## B.3 Local-to-local

After all multipole-to-local, M2L, operations are performed at level 0, there is the L2L from level 0 to 1, as shown in Fig. 5.3(c). The operations at level greater than 0 follows as in free-domain FMM, with the addition of boxes at level  $\ell > 1$  inside the periodic images neighbors to the central domain.