AMERICAN UNIVERSITY OF BEIRUT

TOWARDS RESURRECTING GRID-FREE THREE DIMENSIONAL VORTEX METHODS

by
SAMER OMAR SALLOUM

A dissertation
submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
to the Department of Mechanical Engineering
of Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
February 2021

AMERICAN UNIVERSITY OF BEIRUT

TOWARDS RESURRECTING GRID-FREE THREE DIMENSIONAL VORTEX METHODS

by

SAMER OMAR SALLOUM

Approved by:

Dr. Kamel Abou Ghali, Professor Mechanical Engineering	Committee Head
Dr. Issam Lakkis, Professor Mechanical Engineering	Advisor
Dr. Omar Knio, Professor Applied Mathematics and Computer Science	Member of Committee
Dr. Mu'Tasem Shehadeh, Associate Professor Mechanical Engineering	Member of Committee
Dr. Jihad Touma, Professor	Member of Committee

Physics

Dr. Rio Yokota, Associate Professor — Member of Committee

Global Scientific Information and Computing

Date of thesis defense: February 5, 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	Salloum	Samer	Omar
	Last	First	Middle

○ Master's Thesis ○ Master's Project ◉ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repos-itories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☒ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after:

One ☒ year from the date of submission of my thesis, dissertation or project.
Two ☐ years from the date of submission of my thesis , dissertation or project.
Three ☐ years from the date of submission of my thesis , dissertation or project.

21/4/2021

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

Foremost, I would first like to thank my advisor, Prof. Issam Lakkis, for his invaluable assistance, support, and great motivation throughout my dissertation research. His expertise was invaluable in formulating the research questions and methodology, and it was a great privilege and honor to work and study under his guidance. Besides, I would like to express my deep and sincere gratitude to all committee members for their invaluable comments and discussions. Last but not least, I would like to thank my parents and friends and all those who have contributed to this dissertation and supported me all along the way.

An Abstract of the Thesis of

Samer Omar Salloum for Doctor of Philosophy
Major: Mechanical Engineering

Title: Towards Resurrecting Grid-Free Three Dimensional Vortex Methods

We present expressions for approximating the velocity and vortex stretching vectors induced by a far-field collection of point vortices and we propose a set of recurrence relations to properly evaluate these expressions. Expressed as truncated series of spherical harmonics, these approximations are used to develop an error-controlled hybrid adaptive fast solver that combines both $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ schemes. For a given accuracy, the adaptive solver is used in the context of regularized vortex methods to optimize the speed of the velocity and vortex stretching calculation. This is accomplished by introducing criteria for cell division in building of the tree, conversion of multipole to local expansion coefficients in the downward pass, stopping of the downward pass and choosing between direct and fast summation to compute the vector fields. These criteria are based on key parameters (p, n_F, n_T, d_σ) which take into account the elements distribution, choice of the regularization function, and the computer architecture. The proposed solver automatically adapts to the evolving flow-field by periodically updating the optimal values of these parameters to minimize the speed, while meeting the accuracy constraints, by balancing near and far-field calculations. Accuracy of the introduced expressions is investigated by inspecting the convergence of the velocity and the vortex stretching vectors as a function of the expansion order. Performance of the proposed adaptive scheme is investigated in terms of the dependence of cost and accuracy on the various controlling parameters. Evolution of the optimal values of adaptive solver parameters along with the associated computational savings are presented for the case of collision of two vortex rings over a reasonable time span.
The adaptive solver, along with the introduced expressions, are used to assess the overall performance of three-dimensional grid-free regularized vortex methods by simulating the collision of two vortex rings, over a long period of time, for different values of Reynolds number covering the range $500 - 2000$. These

methods typically rely on operator splitting to handle diffusion and convection separately. In our implementation, the convection step employs a second order Runge-Kutta time integration scheme, where the particles velocities and vortex stretching vectors are computed using the adaptive fast solver that employs the proposed expressions. To model diffusion, we use an extension of the smoothed redistribution scheme to 3D unbounded flows and we compare it with the particle strength exchange diffusion model. We report on the capacity and limitation of the redistribution diffusion model to maintain a divergence free vorticity field without the need to resort to explicit methods and we explore the accuracy of different methods traditionally used to enforce the divergence free condition on the vorticity field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Literature Review

Vortex methods[1, 2, 3, 4] are Lagrangian mesh-free methods for numerically solving the vorticity equation. They were originally based on Helmholtz's circulation theorem which states that, in a barotropic inviscid fluid subject to conservative body forces, the circulation is conserved along particles trajectories. Under these conditions, a complete description of the flow field is obtained by tracking the motion of vorticity-carrying fluid elements[4], starting with an initial vorticity distribution. In these methods, the vorticity field is discretized using singular or regularized vortex blobs. Operator splitting is used to numerically solve the vorticity equation in two separate steps. The convection step commonly employs a higher order Runge-Kutta time integration scheme, where the particles are advected with the local velocity field, and their vorticity strengths are adjusted to account for straining of the vorticity vector by the local velocity gradient field. The velocity vector and the velocity gradient tensor at particles locations are computed using a Biot-Savart type summation over all the particles. In the diffusion step, the vorticity strengths of the particles are modified to account for viscous effects. As the number of elements is increased, regularized vortex methods converge to the exact solution provided that the core overlapping condition is maintained and that particle remeshing is done every few time steps[5]. Without particle remeshing, Lagrangian methods rapidly lose their accuracy especially in high inertia flows where large strain rates cause the particles to move away from each other thereby violating the overlap condition. Particle remeshing usually employs the M_4' high order interpolation scheme which preserves the total circulation, linear and angular impulses. However, this interpolation scheme introduces some dissipation[6], effectively acting as a hyper-viscosity.

Since their inception, vortex methods underwent tremendous progress. The first vortex simulations were conducted by Rosenhead[7] using 12 point vortices in

two dimensions. Birkhof and Fisher[8] showed that the vortices must be regularized to satisfactorily approximate a vortex sheet. Many of the aspects of modern vortex methods are due to the pioneering work of Chorin[9], who introduced the notion of vortex blob and repeated the work of Rosenhead by using regularized vortex cores. Chorin also introduced the random-walk method for grid-free modeling of diffusion in slightly viscous two-dimensional flows. The advent of grid-free diffusion modeling methods[10, 2, 11, 12, 13], along with the development of $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ fast solvers for fast computation of the velocity and vortex stretching vectors[14, 15] and the proper handling of different boundary conditions[16, 17] and of source terms[18, 19] made vortex methods potentially a viable alternative to grid-based methods for numerically solving the Navier-Stokes equations at a high resolution[4]. Vortex methods have been utilized extensively in simulating inviscid and high inertia flows including vortex sheets[20, 21], unsteady separated flows[22, 23, 24], high Reynolds number wakes[25], and various three-dimensional problems involving vortex rings, jets, and wakes[26, 27, 28]. They have also been used to study reacting shear layers[29, 30], Rayleigh Taylor flows[31], fire plumes[32, 33], diffusion-controlled combustion[13], and unsteady two-dimensional compressible flows[34]. These methods were also employed for simulating buoyant flows[35, 19], reacting flows[36, 37], radiation in participating media[38], and reacting-radiating flows[39].

When compared to grid-based methods, grid-free vortex methods possess several desirable features that kept them of continuous interest. In addition to being completely mesh-free, vortex methods are highly adaptive because only regions of non-vanishing vorticity are represented by the computational elements[40]. Furthermore, since these elements are advected by the velocity field, there is no need for explicit numerical discretization of the convection term, which is a common source of numerical diffusion in grid-based methods. These methods, by virtue of their Lagrangian nature, demonstrated exceptional ability to capture the small scale features in high inertia flows. Another advantage is that the continuity equation is automatically satisfied for incompressible fluids since the velocity field is computed as the curl of a vector potential. The absence of the pressure term in the vorticity equation is another attractive feature[3]. Moreover, the satisfactory energy conservation properties of vortex methods, in addition to their small dispersion error, made them suitable to simulate convection-dominated problems[6]. However, and although it was reported in[41] that vortex methods can be faster than Eulerian finite-difference methods due to the increased stability and larger time steps, vortex methods had not yet reached a level of maturity where they can be considered as an efficient alternative to Eulerian methods. This is because vortex methods still face some major challenges, especially the challenge of maintaining a divergence free vorticity field without affecting the conserved quantities. In fact, although the divergence of the initial vorticity can be made small enough by using a sufficiently large number of elements, maintaining the divergence free

condition of the vorticity field remains without any doubt the weakest point of the 3D vortex particle method[42]. According to Winckelmans[5], the vorticity divergence is naturally damped by viscous diffusion especially when the simulation is well-resolved and particle remeshing is used. However, for long time simulations, enforcing the divergence free condition is essential especially in high inertia flows. To this end, Winckelmans[43] proposed the W-scheme and reported, upon examining this scheme in addition to the P-scheme[44], that both schemes are diffusive.

The accurate calculation of the velocity and vortex stretching vectors is crucial in vortex methods. In fact, the velocity is used to advect vortex elements using a time step that is typically larger than that used in Eulerian methods. Thus, advecting with an inaccurate velocity over a large time step will introduce errors in the particles trajectories resulting in an inaccurate vorticity field over short times. Moreover, since the vortex stretching vector, expressed as the dot product of the vorticity vector and the rate of strain tensor, is used to update the vorticity using the Helmholtz equation, inaccurate computation of the rate of strain tensor over a large time step will introduce time integration errors in the vorticity representation and will eventually impact the flow invariants such as total circulation, linear impulse, and angular impulse. Although it is essential to accurately compute the velocity and its spatial derivatives, calculating these terms through direct summation comes at a prohibitive $\mathrm{O}(N^2)$ cost, especially for simulating three dimensional flows, where a large N is needed to capture the small scale features. This challenge has been alleviated by the advent of fast methods for the N body problems[14, 15, 45].
Traditionally, two variants of fast methods for approximating the Biot-Savart summation are found in the literature. They are the vortex in cell methods (VIC)[45] and the hierarchical N-body fast multipoles solver[14, 15]. The vortex in cell methods (VIC) are tailored for computing the velocity and vortex stretching on a grid[46]. In fact, the vorticity is first interpolated on a grid and the vector potential is calculated on this grid using a Poisson solver based on the Fast Fourier Transform. The velocity and vortex stretching are then calculated by centered finite difference on the grid and then interpolated back to the particle locations. The particles are then advected using the velocities at their locations and the vorticity strengths are adjusted to account for vortex stretching. VIC method still has negligible dispersion error since the particles are advected in a Lagrangian way. Moreover, the divergence free condition of the vorticity field is maintained by projecting the vorticity field onto a divergence free basis. This operation provides an efficient and essentially non-diffusive "relaxation" scheme, which allows for simulations at high Reynolds number and for arbitrarily long times[5]. Although VIC methods are very fast and relatively simple to implement, they face the challenge of setting up the proper boundary conditions of the Poisson solver in unbounded or semi-bounded domains. The grid must be taken much larger than the vorticity field region to provide approximate analyt-

ical boundary conditions for the vector potential, $\vec{\psi}$. To this end, Chatelain and Koumoutsakos[47] presented a computationally efficient, adaptive solver for the solution of the Poisson and Helmholtz equation in domains with combinations of unbounded and periodic directions. The method relies on FFTs in all directions and doubles the domain size in the non-periodic directions.
Alternatively, hierarchical N-body fast solvers are flexible and preserve the grid-free nature of vortex methods. The following two variants of these methods have been developed, treecodes[14] and FMMs[48]. Compared to the direct summation, which has a complexity of $O(N^2)$, the treecode algorithm[14] achieves a complexity of $\mathcal{O}(N \log N)$. It does so by clustering source particles into larger bins within a recursive tree-based division operation and using multipole expansions to approximate their influence on the targets. The fast multipole method (FMM)[15] extends the Barnes-Hut treecode[14] by including local expansions to further reduce the computational cost. By clustering the nearby target particles[49], in addition to the source particles, the FMM[15] can achieve $\mathcal{O}(N)$ complexity. Implementation of FMM on de-singularized particles increases the cost to $O(N^{1.2})$[43]. A combination of vortex-in-cell (VIC) and parallel fast multipole methods (FMM) to simulate unbounded and half-bounded incompressible unsteady flows is presented in[6]. The exact Dirichlet boundary conditions for $\vec{\psi}$ are obtained using the parallel fast multipole (FMM) method. These boundary conditions are used to solve the Poisson equation on a grid without the need of expanding the grid dimension.

1.2 Motivation and Objectives

In the last two decades, research in grid-free vortex methods has been too slow. This is because some key challenges hindered the progress of these methods. These challenges, which have not yet been fully overcome, are the reason why vortex methods have not yet caught up with mesh-based methods. These challenges include enforcing a divergence free vorticity field without affecting conserved quantities, handling source term, efficient treatment of vorticies near solid boundaries, and accurate and cost-effective evaluation of the velocity and vortex stretching vectors. However, despite the challenges, vortex methods hold the potential to have advantages over Eulerian methods in some problems.
The main objective of this research is to propel vortex methods forward ploughing through the challenges facing the evolution of three-dimensional grid-free three-dimensional vortex methods to realize the potential that these methods hold. The focus is on truly grid-free methods that do not rely on a grid whatsoever. This excludes vortex in cells methods and implementations of vortex methods that employ regridding. Throughout our effort, we formulated the following questions to address the challenges identified: (i) How can we further increase the speed for a given accuracy measure?, (ii) What is an accurate, cost-effective, grid-free

diffusion model?, (iii) How can we handle vortex stretching in an accurate, cost-effective, and grid-free manner, and (iv) How can we enforce the divergence-free condition on the vorticity vector field?

For vortex methods to be taken seriously from the mainstream CFD community, researchers in vortex methods must tackle these challenges and overcome these difficulties. As such, we outlined our ambitious plan to deal with some of the key challenges of grid-free three dimensional vortex methods as follows:

1. Introduce simplified spherical harmonics-based expressions for the velocity and vortex stretching vectors and properly evaluating these expressions near and at the poles using a set of recurrence relations.

2. Introduce an error-controlled hybrid adaptive fast solver for regularized vortex methods.

3. Investigate the efficiency and limitation of the Redistribution Method to simulate vorticity diffusion in unbounded domain and present a comparison with PSE Scheme.

4. Assess existing methods for enforcing the divergence free condition of the vorticity field.

We believe that this research will have an impact on the grid-free vortex methods field and help resurrect these methods.

1.3 Thesis Outline

The thesis is organized as follows: In chapter 1, we present expressions for approximating the velocity and vortex stretching vectors induced by a far-field collection of point vortices and we propose a set of recurrence relations to properly evaluate these expressions. In chapter 2, we introduce an error-controlled hybrid adaptive fast solver along with the key parameters employed in tree division and decision making for load balancing. We also evaluate the key parameters in terms of the order of expansion using cost analysis and time measurements specific to computer architecture, code implementation, choice of regularization function, and identity of the vector quantity being evaluated. Decomposition of the error and derivation of upper bounds of the constituting components are also presented in this chapter. In chapter 3, we present an extension to three dimensions of the smoothed redistribution method. This grid -free diffusion model is used throughout this paper to simulate three dimensional flows. Basic formulation of the particle strength exchange diffusion model is also presented in this chapter. In chapter 4 we assess existing methods for enforcing the divergence free condition

of the vorticity field. In Chapter 5, extensive validation of the introduced velocity and vortex stretching expressions and the proposed error-controlled adaptive solver is performed. Moreover, the overall performance of three dimensional grid-free vortex methods is assessed by simulating the collision of two vortex rings, over a long period of time, for different values of Reynolds number. In Chapter 6, we present a summary of conclusions and recommends future work that can be done.

Chapter 2

Proper Evaluation of Spherical Harmonics-Based Expressions For The Velocity and Vortex Stretching Vectors in Three-Dimensional Grid-Free Vortex Methods

To accurately simulate three dimensional flows, a very large number of elements is required to resolve the various length scales. Direct evaluation of the velocity and vortex stretching vectors requires $O(N^2)$ computational operations of all the pairwise interaction in a system of N elements. The prohibitive cost of the direct evaluation renders the fast multipole method (FMM) an essential tool to execute particle-based simulations in a reasonable amount of time. Yet, little effort has been devoted to formulate compact expressions of the velocity and vortex stretching vectors as a truncated series of spherical harmonics. Winckelmans[43], Ploumhans[50], and Cocle[6], among others used analytical derivatives of the multipole expansions to evaluate the velocity vector and the velocity gradient. Yokota[51, 52, 53] described the expressions for both the velocity and stretching vectors in several of his papers but he didn't discuss how to deal with the spatial derivatives of spherical harmonics. Berdowski (2016)[54] presented a detailed derivation for approximating the velocity and stretching vectors using multipole expansions and implemented them in $\mathcal{O}(N \log N)$ fast multipole algorithm. The discussion, however, does not include local expansions since they are not needed in the $\mathcal{O}(N \log N)$-type fast solver. Furthermore, evaluating the velocity and vortex stretching vectors using the equations presented in[54] requires dividing by $sin\theta$ and $sin^2\theta$, where direct numerical evaluation is problematic as the polar

angle θ approaches zero at the poles.
In this chapter, we present simple alternative expressions for the velocity and vortex stretching vectors using both local and multipole expansion coefficients, and we propose a set of recurrence relations to properly evaluate these expressions over the entire range of the polar angle.

2.1 Vortex Methods and the Vorticity Transport Equation

For three dimensional incompressibe flow, $\nabla.\vec{u} = 0$, the evolution of the vorticity, $\vec{\omega}$, is described by the Helmholtz's vorticity equation:

$$\frac{d\vec{\omega}}{dt} = (\vec{\omega}.\nabla)\vec{u} + \frac{\nabla\rho \times \nabla p}{\rho} + \nabla^2\vec{\omega} \tag{2.1}$$

where t is time, $\vec{u}$ is the velocity, ρ is the density, p is the pressure, and d/dt is the Lagrangian derivative $d/dt = \partial/\partial t + \vec{u} \cdot \nabla$.
For an incompressible flow in unbounded domain, the velocity vector field is computed as the curl of a vector potential field $\vec{\psi}$

$$\vec{u} = \nabla \times \vec{\psi} \tag{2.2}$$

Noting that the vorticity vector field is the curl of the velocity vector field, we obtain:

$$\vec{\omega} = \nabla \times \vec{u} = \nabla \times (\nabla \times \vec{\psi}) = -\nabla^2\vec{\psi} + \nabla(\nabla.\vec{\psi}) \tag{2.3}$$

For $\nabla.\vec{\psi} = 0$, the vector potential field, $\vec{\psi}$, is governed by the Poisson equation

$$\nabla^2\vec{\psi} = -\vec{\omega} \tag{2.4}$$

The Green's function for the Poisson equation (2.4) in an unbounded domain is $G(\vec{x}) = -\frac{1}{4\pi\|\vec{x}\|}$, so that the vector potential field, $\vec{\psi}$, is calculated as:

$$\vec{\psi}(\vec{x}, t) = \frac{1}{4\pi}\int \frac{\vec{\omega}(\vec{y}, t)}{\|\vec{x} - \vec{y}\|} d^3y \tag{2.5}$$

In vortex methods, the vorticity field is discretized using vortex elements. The elements can be point vortices (singular representation of the vorticity field) or blobs (regularized representation of the vorticity field). Associated with each element i are the position vector $\vec{x}_i$ and the strength vector $\vec{\alpha}_i$. Assuming a singular representation of the vorticity field,

$$\vec{\omega}_\delta(\vec{x}, t) = \sum_{i=1}^{N} \vec{\alpha}_i \delta(\vec{x} - \vec{x}_i), \tag{2.6}$$

where N is the number of elements and δ is the Dirac delta function, the vector potential field and the velocity vector field can be approximated as

$$\vec{\psi}_\delta(\vec{x},t) = \frac{1}{4\pi}\sum_{i=1}^{N}\frac{\vec{\alpha}_i(t)}{\|\vec{x}(t)-\vec{x}_i(t)\|} \tag{2.7}$$

$$\vec{u}_\delta(\vec{x},t) = \nabla\times\vec{\psi}_\delta = -\frac{1}{4\pi}\sum_{i=1}^{N}\frac{(\vec{x}(t)-\vec{x}_i(t))}{\|\vec{x}(t)-\vec{x}_i(t)\|^3}\times\vec{\alpha}_i(t) \tag{2.8}$$

A regularized representation of the vorticity field may be expressed as

$$\vec{\omega}_\sigma(\vec{x},t) = \sum_{i=1}^{N}\vec{\alpha}_i(t)\zeta_\sigma(\vec{x}-\vec{x}_i), \tag{2.9}$$

where ζ_σ is a regularization function and σ is the associated smoothing radius. The regularization function is usually taken as a radially symmetric function, $\zeta_\sigma(\vec{x}-\vec{x}_i) = \frac{1}{\sigma^3}\zeta\left(\frac{\|\vec{x}-\vec{x}_i\|}{\sigma}\right) = \frac{1}{\sigma^3}\zeta(\frac{r}{\sigma})$, that satisfies the normalization condition

$$4\pi\int_0^\infty \zeta_\sigma(\rho)\rho^2 d\rho = 1. \tag{2.10}$$

A regularization function of order l, that conserves up to order $l-1$ moments as the Dirac function, can be constructed as described in[55]. Using a radially symmetric regularization function, the corresponding vector potential field and the velocity vector field can be approximated as

$$\vec{\psi}_\sigma(\vec{x},t) = \sum_{i=1}^{N} G_\sigma(\vec{x}-\vec{x}_i)\vec{\alpha}_i(t) \tag{2.11}$$

$$\vec{u}_\sigma(\vec{x},t) = \nabla\times\vec{\psi}_\sigma = -\sum_{i=1}^{N} q_\sigma\left(\vec{x}(t)-\vec{x}_i(t)\right)\frac{(\vec{x}(t)-\vec{x}_i(t))}{\|\vec{x}(t)-\vec{x}_i(t)\|^3}\times\vec{\alpha}_i(t) \tag{2.12}$$

Where G solves the Poisson equation $\nabla^2 G = -\zeta(r)$, $G_\sigma(r) = \frac{1}{\sigma}G(\frac{r}{\sigma})$, $q(r) = \int_0^r \zeta(\rho)\rho^2 d\rho$, and $q_\sigma(r) = q(\frac{r}{\sigma})$[56].

Direct evaluation of the velocity vector and the rate of strain tensor requires $O(N^2)$ computational operations of all the pairwise interaction in a system of N particles. The high cost of direct evaluation severely compromises the viability of the method to simulate transient flows, especially for cases where the number of particles, required to accurately resolve the vorticity field, grows considerably with time. Alternatively, the fast multipole method (FMM) enables conducting these simulations with a reasonable cost. Next, we briefly present the $\mathcal{O}(N\log N)$ and $\mathcal{O}(N)$ fast solvers.

2.2 The Fast Multipole in Three Dimensions

The $\mathcal{O}(N \log N)$ fast multipole scheme is based on approximating, to an arbitrary precision, the vector potential induced by clusters of singular vortices in the far field using multipole expansions[57]. Implementation of $\mathcal{O}(N)$ scheme requires, in addition, the use of local expansions to further reduce the order of the computations. These two expansions are based on spherical harmonics, $Y_n^m(\theta, \varphi), n \in \mathbb{N}, |m| \leq n$, which form a frequency-space basis for representing functions on a sphere. Thus, any spherical vector field $\vec{\phi}(\theta, \varphi)$ may be expanded as the linear combination $\vec{\phi}(\theta, \varphi) = \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} \vec{C}_n^m Y_n^m(\theta, \varphi)$, where the coefficients $\vec{C}_n^m$ are computed by projecting $\vec{\varphi}$ onto each basis function Y_n^m.

We note that for a regularized representation of the vorticity field, these same expansions (which are based on point vortices) are used to calculate the effect of the far field. In this case, the use of multipole expansion coefficients to approximate the far-field vector potential $\vec{\psi}_\sigma$ will induce an error that is made up of the following two components:

1. A component resulting from truncating multipole expansions at some order p: This error is related to the series expansion used to represent the kernel $\frac{1}{r}$ at large distance and is a function of the order of expansion p and the Multipole Acceptance Criteria. This error decreases exponentially as p increases and converges to zero as $p \to \infty$.

2. A component resulting from approximating the Biot-Savart kernel by a $\frac{1}{r}$ kernel: This error arises due to the fact that G_σ deviates from $\frac{1}{4\pi r}$. This error mainly depends on the size of the smallest boxes (at the deepest level of the tree). In order to neglect this error, we must keep the leaf boxes width above a minimum multiple of the core function smoothing radius. In three dimensional flows, this will pose a considerable computational overload on the FM schemes, since it will dramatically increase the particle to particle interaction.

2.2.1 Multipole Expansion of the Vector Potential Field

Suppose that s vortices having strengths $(\vec{\alpha}_j, j = 1...s)$ are located at the points $\vec{Q}_j = (\rho_j, \theta_j, \varphi_j)$ inside the sphere D_Q, with $|\rho_j| < a$ (see Figure 2.1, left), then at any $\vec{P} = (r, \theta, \varphi)$ with $r > a$, the vector potential $\vec{\psi}_\delta$, is approximated by the following multipole expansion[15]

$$\vec{\psi}_\delta(\vec{P}, t) = \frac{1}{4\pi} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{\vec{M}_n^m}{r^{n+1}} Y_n^m(\theta, \varphi) \tag{2.13}$$

with

$$\vec{M}_n^m = \sum_{j=1}^{s} \vec{\alpha}_j \rho_j^n Y_n^{-m}(\theta_j, \varphi_j) \tag{2.14}$$

Furthermore, for any $p \geq 1$

$$\|\vec{\psi}_\delta(\vec{P}, t) - \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{\vec{M}_n^m}{r^{n+1}} Y_n^m(\theta, \varphi)\| \leq \frac{1}{4\pi} \frac{\Gamma_{D_Q}}{r-a} \left(\frac{a}{r}\right)^{p+1} \tag{2.15}$$

where

$$\Gamma_{D_Q} = \sum_{j=1}^{s} \| \vec{\Gamma}_j \| \tag{2.16}$$

Thus, at a given point $P = (r, \theta, \varphi)$ with $r > a$, the vector potential is approximated by

$$\vec{\psi}_\delta(\vec{P}, t) \simeq \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{\vec{M}_n^m}{r^{n+1}} Y_n^m(\theta, \varphi) \tag{2.17}$$

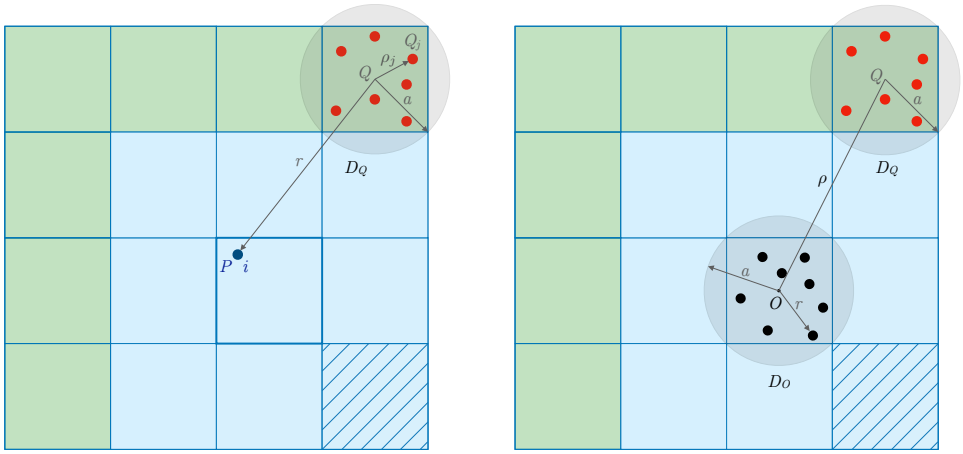

Figure 2.1: Schematic for the multipole and local expansions.

2.2.2 Local Expansion of the Vector Potential Field

Suppose that s vortices having strengths $(\vec{\alpha}_j, j = 1...s)$ are located inside the sphere D_Q of radius a with center at $Q = (\rho, \alpha, \beta)$ (see Figure 2.1, right), and that $\rho = (c+1)a$ with $c > 1$, then for any point $P(r, \theta, \phi)$ inside D_0 of radius a centered at the origin, the vector potential due to vortices $(\vec{\alpha}_j, j = 1...s)$ inside D_Q is described by the following local expansion[15]

$$\vec{\psi}_\delta(P, t) = \frac{1}{4\pi} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \vec{L}_n^m Y_n^m(\theta, \varphi) r^n, \tag{2.18}$$

where

$$\vec{L}_n^m = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} \frac{\vec{M}_j^k . i^{|m-k|-|m|-|k|} . A_j^k . A_n^m . \rho^j . Y_{n+j}^{k-m}(\alpha, \beta)}{(-1)^j A_{n+j}^{k-m} . \rho^{n+j+1}}, \tag{2.19}$$

with A_n^m defined by

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)!(n+m)!}} \tag{2.20}$$

Furthermore, for any $p \geq 1$,

$$\| \vec{\psi}_\delta(P,t) - \frac{1}{4\pi} \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \vec{L}_n^m Y_n^m(\theta,\varphi) r^n \| \leq \frac{\Gamma_{D_Q}}{a(c-1)} \left(\frac{1}{c}\right)^{p+1} \tag{2.21}$$

Thus, at a given point $P = (r, \theta, \varphi)$ with $r < a$, the vector potential due to vortices inside D_Q is approximated by

$$\vec{\psi}_\delta(P,t) = \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \vec{L}_n^m Y_n^m(\theta,\varphi) r^n \tag{2.22}$$

The spherical harmonics can be expressed in term of the associated Legendre polynomials[15] as

$$Y_n^m(\theta,\varphi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta)\, e^{im\varphi} \tag{2.23}$$

There are many ways to define the associated Legendre polynomials P_n^m and its derivatives[58]. We propose next a set of recurrence relations to properly evaluate these polynomials and their first and second order derivatives over the entire range of polar angle.

2.3 Alternative Simple Expressions for the Velocity and Vortex Stretching Vectors Expressed as a Truncated Series of Spherical Harmonics

In three-dimensional vortex methods, computing the velocity and the vortex stretching vectors at the particles positions is required every time step. Expressions for the far-field component of these two vectors in term of spherical harmonics can be obtained respectively from the corresponding spatial derivatives of the vector potential approximations in Eqs. (2.17) and (2.22), as presented below. These expressions are used in the context of $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ FM schemes to reduce the computational cost of the advection step.

2.3.1 Expressions for the Velocity Vector in Term of Multipole and Local Expansion Coefficients

The velocity vector $\vec{u}_\delta$ is computed as the curl of the vector potential $\vec{\psi}_\delta$. Expanding $\vec{\psi}_\delta$ in term of multipole expansion coefficients $\vec{M}_n^m$ using Eq. (2.17), and taking the curl of $\vec{\psi}_\delta$, the velocity vector can be expressed as

$$\vec{u}_\delta = \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \nabla \left(\frac{Y_n^m(\theta, \varphi)}{r^{n+1}} \right) \times \vec{M}_n^m \tag{2.24}$$

In Cartesian coordinates, we get

$$\vec{u}_\delta = \frac{1}{4\pi} \sum_{n=1}^{p} \sum_{m=-n}^{n} \frac{C_n^m e^{im\varphi}}{r^{n+2}} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} -(n+1)P_n^m \\ \frac{dP_n^m}{d\theta} \\ \frac{imP_n^m}{\sin\theta} \end{pmatrix} \times \vec{M}_n^m, \tag{2.25}$$

where

$$C_n^m = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \tag{2.26}$$

Expanding $\vec{\psi}_\delta$ in term of local expansion coefficients $\vec{L}_n^m$ using equation (2.18), the velocity vector can be expressed as

$$\vec{u}_\delta = \nabla \times \vec{\psi}_\delta(P,t) = \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \nabla(r^n Y_n^m(\theta, \varphi)) \times \vec{L}_n^m \tag{2.27}$$

In cartesian coordinates, we get

$$\vec{u}_\delta = \frac{1}{4\pi} \sum_{n=1}^{p} \sum_{m=-n}^{n} r^{n-1} C_n^m e^{im\varphi} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} nP_n^m \\ \frac{dP_n^m}{d\theta} \\ \frac{imP_n^m}{\sin\theta} \end{pmatrix} \times \vec{L}_n^m \tag{2.28}$$

2.3.2 Expressions for the Vortex Stretching Vector in Term of Multipole and Local Expansion Coefficients

The vortex stretching vector, $(\vec{\alpha}.\nabla)\vec{u}_\delta$, quantifies the stretching and tilting of the vorticity due to the flow velocity gradients. It is computed as the projection of the rate of strain tensor (gradient of the velocity vector) along the direction of the vorticity vector $\vec{\omega}$, scaled by the magnitude of the vorticity $||\vec{\omega}||$.

The rate of stain tensor is expressed in Cartesian coordinates as

$$\nabla \vec{u}_\delta = \begin{pmatrix} \frac{\partial u_\delta}{\partial x} & \frac{\partial u_\delta}{\partial y} & \frac{\partial u_\delta}{\partial z} \\ \frac{\partial v_\delta}{\partial x} & \frac{\partial v_\delta}{\partial y} & \frac{\partial v_\delta}{\partial z} \\ \frac{\partial w_\delta}{\partial x} & \frac{\partial w_\delta}{\partial y} & \frac{\partial w_\delta}{\partial z} \end{pmatrix} \tag{2.29}$$

Thus the rate of stain tensor is determined completely by the three vectors $\frac{\partial \vec{u}_\delta}{\partial x}$, $\frac{\partial \vec{u}_\delta}{\partial y}$, $\frac{\partial \vec{u}_\delta}{\partial z}$, representing the variation of the velocity vector along the x, y, and z directions respectively. Applying the chain rule in partial differentiation we obtain:

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \vec{u}_\delta = \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial r} \\ \frac{1}{r}\frac{\partial}{\partial \theta} \\ \frac{1}{r\sin\theta}\frac{\partial}{\partial \varphi} \end{pmatrix} \vec{u}_\delta \tag{2.30}$$

Expanding $\vec{u}_\delta$ in term of multipole expansion coefficients $\vec{M}_n^m$ using Eq. (2.25) and differentiating with respect to r, θ, and ϕ, we obtain:

$$\frac{\partial \vec{u}_\delta}{\partial r} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} \frac{-(n+2)}{r^{n+3}} C_n^m e^{im\varphi} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} -(n+1)P_n^m \\ \frac{dP_n^m}{d\theta} \\ \frac{imP_n^m}{\sin\theta} \end{pmatrix} \times \vec{M}_n^m \tag{2.31}$$

$$\frac{1}{r}\frac{\partial \vec{u}_\delta}{\partial \theta} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} \frac{1}{r^{n+3}} C_n^m e^{im\varphi} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} -(n+2)\frac{dP_n^m}{d\theta} \\ \frac{d^2P_n^m}{d\theta^2} - (n+1)P_n^m \\ i\frac{d}{d\theta}(\frac{mP_n^m}{sin\theta}) \end{pmatrix} \times \vec{M}_n^m \tag{2.32}$$

$$\frac{1}{r\sin\theta}\frac{\partial \vec{u}_\delta}{\partial \varphi} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} \frac{1}{r^{n+3}} C_n^m e^{im\varphi} \begin{pmatrix} \sin\varphi D_n^m - im\cos\varphi D_n^m - (m^2-1)\frac{P_n^m}{\sin^2\theta}) \\ -\cos\varphi D_n^m - im\sin\varphi D_n^m - (m^2-1)\frac{P_n^m}{\sin^2\theta}) \\ -im(\frac{dP_n^m}{\partial\theta} + (n+1)\frac{\cos\theta}{\sin\theta}P_n^m) \end{pmatrix} \times \vec{M}_n^m \tag{2.33}$$

where

$$D_n^m = \frac{d^2P_n^m}{d\theta^2} + (n+1)^2 P_n^m \tag{2.34}$$

Expanding $\vec{u}_\delta$ in term of local expansion coefficients $\vec{L}_n^m$ using Eq. (2.28) and differentiating with respect to r, θ, and ϕ, we obtain:

$$\frac{\partial \vec{u}_\delta}{\partial r} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} (n-1) r^{n-2} C_n^m e^{im\varphi} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} nP_n^m \\ \frac{dP_n^m}{d\theta} \\ \frac{imP_n^m}{\sin\theta} \end{pmatrix} \times \vec{L}_n^m \tag{2.35}$$

$$\frac{1}{r}\frac{\partial \vec{u}_\delta}{\partial \theta} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} r^{n-2} C_n^m e^{im\varphi} \begin{pmatrix} \sin\theta\cos\varphi & \cos\theta\cos\varphi & -\sin\varphi \\ \sin\theta\sin\varphi & \cos\theta\sin\varphi & \cos\varphi \\ \cos\theta & -\sin\theta & 0 \end{pmatrix} \begin{pmatrix} (n-1)\frac{dP_n^m}{d\theta} \\ \frac{d^2P_n^m}{d\theta^2} + nP_n^m \\ i\frac{d}{d\theta}(\frac{mP_n^m}{sin\theta}) \end{pmatrix} \times \vec{L}_n^m \tag{2.36}$$

$$\frac{1}{r\sin\theta}\frac{\partial \vec{u}_\delta}{\partial \varphi} = \frac{1}{4\pi} \sum_{n=2}^{p} \sum_{m=-n}^{n} r^{n-2} C_n^m e^{im\varphi} \begin{pmatrix} \sin\varphi F_n^m - im\cos\varphi F_n^m - (m^2-1)\frac{P_n^m}{\sin^2\theta} \\ -\cos\varphi F_n^m - im\sin\varphi F_n^m - (m^2-1)\frac{P_n^m}{\sin^2\theta} \\ -im(\frac{dP_n^m}{d\theta} - n\frac{\cos\theta}{\sin\theta}P_n^m) \end{pmatrix} \times \vec{L}_n^m \tag{2.37}$$

where

$$F_n^m = \frac{d^2P_n^m}{d\theta^2} + n^2 P_n^m \tag{2.38}$$

2.3.3 Proper Evaluation of the Velocity and Vortex Stretching Vectors Over the Entire Range of the Polar angle

So far, we derived expressions for the velocity and vortex stretching in terms of (P_n^m, $\frac{dP_n^m}{d\theta}$, $\frac{d^2P_n^m}{d^2\theta}$, $\frac{P_n^m}{\sin\theta}$, $\frac{d}{d\theta}\frac{mP_n^m}{\sin\theta}$, and $\frac{P_n^m}{sin^2\theta}$). Direct numerical evaluation of the last three terms yield values that tend to infinity when approaching the poles, owing to the fact that $\sin\theta$ and $\sin^2\theta$ tend to zero as θ approach zero. Next we present a method, based on recurrence relations, that allows accurate evaluation of these terms over the entire range of the polar angle, θ.

Noting that the associated Legendre polynomial can be written in the form $P_n^m = f(\cos\theta)\sin^m\theta$, we introduce the two functions, $\alpha_n^m = \frac{P_n^m}{\sin^2\theta}$ for $m \geq 2$ and $\beta_n^m = \frac{P_n^m}{sin\theta}$ for $m \geq 1$. In Eqs. (2.25) and (2.28) of the velocity vector, and in Eqs. (2.31-2.33) and (2.35-2.37) of the velocity spatial derivatives, the term $\frac{P_n^m}{\sin\theta}$ is always multiplied by m. So we extend the definition of β_n^m by imposing $\beta_n^0 = 0$. Furthermore, the term $\frac{P_n^m}{sin^2\theta}$ is always multiplied by $m(m^2-1)$. We also extend the definition of α_n^m by imposing $\alpha_n^0 = \alpha_n^1 = 0$. By introducing these two functions, we have $m(m^2-1)\alpha_n^m = m(m^2-1)\frac{P_n^m}{\sin^2\theta}$ and $m\beta_n^m = m\frac{P_n^m}{sin\theta}$ for any pair (n, m).

To numerically evaluate these associate Legendre polynomial expressions in a robust manner, we propose an algorithm that uses the following recurrence relations defined for $n > 2$ and $m > 1$.

$$P_n^m = \sin^2\theta\, \alpha_n^m \tag{2.39}$$
$$P_n^1 = \sin\theta\, \beta_n^1 \tag{2.40}$$
$$nP_n^0 = (2n-1)\cos\theta\, P_{n-1}^0 - (n-1)P_{n-2}^0 \tag{2.41}$$

where

$$\alpha_n^n = (-1)^n(2n-1)!!\sin^{n-2}\theta \tag{2.42}$$
$$(n-m)\alpha_n^m = (2n-1)\cos\theta\, \alpha_{n-1}^m - (n+m-1)\alpha_{n-2}^m \tag{2.43}$$
$$\beta_n^m = \sin\theta\, \alpha_n^m \tag{2.44}$$
$$(n-1)\beta_n^1 = (2n-1)\cos\theta\, \beta_{n-1}^1 - n\beta_{n-2}^1 \tag{2.45}$$

and the double factorial, !!, is a factorial where only the even terms are included.

The first order derivative of P_n^m is calculated using the following recurrence relations defined for $n > 1$ and $m > 0$

$$\frac{dP_n^0}{d\theta} = P_n^1 \tag{2.46}$$
$$\frac{dP_n^m}{d\theta} = n\cos\theta\, \beta_n^m - (n+m)\beta_{n-1}^m \tag{2.47}$$

The second order derivative of P_n^m is calculated using the following recurrence relations

$$\frac{d^2P_n^0}{d\theta^2} = \frac{dP_n^1}{d\theta} = n\cos\theta\, \beta_n^1 - (n+1)\beta_{n-1}^1 \tag{2.48}$$
$$\frac{d^2P_n^m}{d\theta^2} = \frac{1}{2}\left[\frac{dP_n^{m+1}}{d\theta} - (n+m)(n-m+1)\frac{dP_n^{m-1}}{d\theta}\right] \tag{2.49}$$
$$\frac{d^2P_n^n}{d\theta^2} = -n\frac{dP_n^{n-1}}{d\theta} = -n\cos\theta\, \beta_n^{n-1} - (2n-1)\beta_{n-1}^{n-1} \tag{2.50}$$

According to Petrovskaya and Vershkov[59], the derivative of $\frac{P_n^m}{\sin\theta}$ can be calculated by the following recurrence relations

$$\begin{aligned}\frac{d}{d\theta}\left(\frac{mP_n^m}{\sin\theta}\right) &= m\left[\frac{1}{\sin\theta}\frac{dP_n^m}{d\theta} - \frac{\cos\theta}{\sin^2\theta}P_n^m\right] \\ &= \frac{1}{2}\left[(m+1)\frac{P_n^{m+1}}{\sin\theta} - (m-1)(n+m)(n-m+1)\frac{P_n^{m-1}}{\sin\theta}\right]\end{aligned} \tag{2.51}$$
$$\frac{d}{d\theta}\left(\frac{mP_n^m}{\sin\theta}\right) = \frac{1}{2}\left[(m+1)\beta_n^{m+1} - (m-1)(n+m)(n-m+1)\beta_n^{m-1}\right] \tag{2.52}$$

For the sake of completeness, and since the spherical harmonics Y_n^m contain the term $e^{im\phi}$, we propose to use Chebyshev's recurrence relations to calculate $\cos(m\phi)$ and $\sin(m\phi)$

$$\cos(m\phi) = 2\cos(\phi)\cos(m-1)\phi - \cos(m-2)\phi \tag{2.53}$$
$$\sin(m\phi) = 2\cos(\phi)\sin(m-1)\phi - \sin(m-2)\phi \tag{2.54}$$

We observed that computing these terms using the above recurrence relations led to considerable savings in CPU time.

Chapter 3

Error-Controlled Hybrid Adaptive Fast Solver for Regularized Vortex Methods

In vortex methods, computing the velocity and vortex stretching vectors at a given position is done by summing the contributions of all vortex elements in the domain. Thus direct evaluation of these vectors at the positions of N elements requires $O(N^2)$ computational operations. The prohibitive cost of large N simulations[49] led to the development of approximate solutions for N body problems.

Two variants of hierarchical N-body methods have been developed, Treecodes and Fast Multipoles Methods (FMMs). To reduce the complexity to $\mathcal{O}(N \log N)$, the Treecode algorithm[14] clusters source elements into progressively larger bins and then approximate their contributions at the targets locations using multipole expansions. In addition to clustering source elements, the FMM[15] reduces the complexity to $\mathcal{O}(N)$ by clustering the nearby target elements as well. The $\mathcal{O}(N)$ FMM scheme then employs local expansions to approximate the contributions of the source elements at the targets locations[49]. These two methods followed distinct paths of evolution and differ from each other in the following ways[60, 61, 49, 62, 48]. (i) For the far field, Treecodes perform cell-element interactions resulting in $\mathcal{O}(N \log N)$ complexity, while FMMs perform cell-cell interactions resulting in $\mathcal{O}(N)$ complexity. (ii) Treecodes conventionally use hard coded Cartesian expansions with a fixed order of expansion (e.g. $p = 3$), while FMMs employ spherical harmonic expansions of an arbitrary order. (iii) To build the interaction list, Treecodes utilize the ratio, θ (also known as the multipole acceptance criterion (MAC)), of the cells size to the distance between cells to decide whether or not to further subdivide a cell. In contrast, FMMs use relationships between the parent, child, and neighbor to build the interaction list[49]. (iv) In Treecodes, the accuracy is controlled by the value of θ (MAC), while in FMMs, the order of expansion p is the controlling parameter. (v) In Treecodes, the expansions are around the centers of mass of the cells, while in FMMs, they are around at the geometrical centers[61]. (vi) While Treecodes are adaptive in nature and permit interaction of cells at different levels, FMMs usually employ a non-adaptive structure of hierarchical grids where only

interactions of cells at the same level are considered[60].

There have been several efforts to take the best of these two methods and to develop optimal algorithms[62].
Cheng et al.[64] extended the work of Greengard[15] to allow for adaptive interaction of cells at different levels in the tree and introduced a mechanism for choosing between cell-cell, cell-element, and element-element interaction. To decide on the type of interaction to be executed, Cheng associated with each cell four different lists of cells, which are assembled as the oct-tree adapts to the evolving particles distribution. Element-element interactions are carried out if the number of elements in a given source cell is less than p^2. Although this criterion captures the expected dependence on p, it is based on a non-optimal value of the critical number of elements in a cell below which direct interaction should be carried out. Moreover, the Oct-tree depth is controlled by the maximum number of elements per cell, N_c, which is specified by the user. Choosing a large value of N_c results in a shallow tree structure where direct interaction with the neighboring cells will likely yield significant increase in the computational cost. On the other hand, if N_c is too small, the scheme will manage to balance cell-cell and element-element interactions, but it will incur the additional cost of evaluating multipole and local expansions coefficients at the centers of cells of high level, although these coefficients are not needed to evaluate the approximate field. In addition, Cheng et al. did not discuss the limitations arising from the use of a regularization function and the dependence on the implementation in different hardware architectures.
Walter Dehnen[60, 61] extended the Treecode method of Barnes and hut[14] for fast evaluation of gravitational forces by including mutual cell-cell interactions, dual tree traversal, and a mass-dependent error-controlled multipole acceptance criteria based on the upper-bound expression of the Newtonian forces. To evaluate cell-cell interactions, Dehnen uses the expressions presented in [63], which were derived from Taylor series expansion in Cartesian coordinates. Dehnen[60] also introduced five different numbers, namely, N_{cb}^{pre}, N_{cb}^{post}, N_{cc}^{pre}, N_{cc}^{post}, N_{cs}. While N_{cs} is used to decide whether or not to divide a cell to carry out Cell-self interactions, the remaining four numbers are used to determine the type of interaction to perform between distinct cells, choosing between cell-cell, cell-body, and body-body interaction. The author determined, through experiments, the optimal values of these five numbers that would result in the most efficient code for a given accuracy and given hardware architecture, but he did not present an automated algorithm for adapting to different hardware architecture or desired level of accuracy. Cell-cell and cell-body interactions are approximated using a Taylor series expansion with a fixed order $p = 3$, so that the MAC number is the only accuracy-control parameter. Upon comparing the performance of his code with that of the 3D adaptive FMM code by Cheng et al.[64], Dehnen[60] showed that, in the low accuracy regime, his code is more than ten times faster and twice as accurate, which makes it favorable for applications where low accuracy is traded for speed (cost), such as collisionless stellar dynamics. This is in agreement with earlier findings that report on the performance of traditional FMM in these low accuracy applications[65]. For high accuracy (e.g. $O(10^{-6})$), however, Cheng et al. FMM outperforms Dehnen's code in terms of speed. According to Dehnen, this behaviour is attributed to the fact

that, while his code increases accuracy by decreasing the MAC number θ (p is always constant), FMM increases the accuracy by increasing p. He concluded by pointing out that a code for which p and θ can be adapted simultaneously would be superior to both codes.
Yokota and Barba[66] extented Dehnen's method by implementing a hybrid Treecode-FMM algorithm that offers control over both the order of expansion and MAC. Depending on the required accuracy, their code switches between Cartesian expansions (lower accuracy) and spherical harmonic expansions (higher accuracy). In addition, their algorithm is capable of auto-tuning the kernels on heterogeneous architectures. The key for auto-tuning is to time all the kernels. The algorithm uses this information to optimally choose the type of interaction to execute in order to evaluate the mutual contribution between non-neighboring cells during the dual tree traversal. However, Yokota did not introduce any mechanism to minimize the computational cost of calculating the interaction between neighboring cells (including cell-self interactions). In fact, the interaction between neighboring cells is performed through direct summation if both cells are at the terminal tree level, that is the number of elements in these terminal cells is less than N_c where the value of N_c is intoduced by the user. A non-optimal value of N_c limits the efficiency of the method. Choosing N_c to be too small results in the additional cost of building the tree and computing the expansion coefficients for a deeper tree. On the other hand, choosing N_c to be too large increases the cost due to the associated element-element interactions. Moreover, Yokota and Baraba did not address the use of a regularization core function and the additional cost incurred to control the regularization error and to maintain a certain level of accuracy.
Yokota and Barba tested their code by comparing its performance to a pure FMM code while fixing the maximum number of elements per leaf, N_c. For a well chosen value of N_c ($N_c = 200$), the results show that the hybrid method always favors cell-cell interactions and, as such, does not offer a clear advantage over the pure FMM. For $N_c = 50$, while the FMM suffers from load imbalance between the far and and near-field calculations, the hybrid method performs much better because it can choose to carry out element-element interactions even if the cell is not a leaf. Yokota and Barba concluded that their hybrid method yields better results due to the automatic fine tuning of the balance between cell-cell, cell-element, and element-element interactions throughout the adaptive tree. The hybrid method removes the wavy behavior of the dependence of the CPU time on N in pure FMMs[66] and achieves optimum performance for all N.
Yokota and Barba also investigated the effect of changing both the multiple acceptance criteria θ and the order of expansion p to achieve maximum performance for a given accuracy on both CPUs and GPUs[62]. They showed that it is inefficient to increase θ to a value that is larger than that of standard FMM codes ($\theta = 1/2$), for both CPU and GPU implementations. Therefore, when designing a hybrid Treecode-FMM algorithm there is no need for a variable MAC since its optimal value is always $1/2$. In this case, the hybrid scheme, developed by Yokota and Barba, yields a similar interaction list as the algorithm proposed by Cheng et al. in [64]. The implementation of the hybrid scheme is, however, much simpler and the algorithm is more general and flexible.

As for ensuring a desired accuracy,we emphazise the work of Winckelmans[43], who

used the Treecode algorithm to simulate viscous and inviscid three dimensional flows using the regularized vortex particle method. In his algorithm, Winckelmans controlled the error incurred in approximating the velocity vector by a truncated Taylor series expansions of a fixed order $p = 2$. For a desired accuracy, Winckelmans used the upper bound expression of the error of the velocity field to evaluate a critical value of the distance between the source cell and target element, d_{crit}, above which the multipole expansion approximation is used. We note that, using an upper bound of the error in the velocity field calculation incurs an additional computational cost, which may be excessively large when compared to using an accurate measure of the error. Moreover, and in order to minimize the regularization error, Winckelmans employed a cutoff distance equal to 5σ below which only direct summation is used, where σ is the smoothing radius of the regularization function.

In this Chapter, we build on the work of Greengard[15] to introduce an error-controlled hybrid method that automatically adapts to the time evolution of the particles distribution to minimize the cost of computing the velocity vector and/or rate of strain tensor fields, while meeting a pre-specified accuracy constraint. The proposed algorithm accomplishes this goal by automatically carrying out periodic load-balancing of the cell-cell, cell-element, and element-element computations as the flow-field evolves in time. To realize optimal load, we introduce various criteria to decide on the order of expansion, tree depth, and whether to compute multipole expansions, local expansions or further divide a cell in the downward and upward tree passes. The criteria are based on key control parameters, whose optimal values not only depend on the evolving elements distribution, but also on the choice of the core function, on the computer architecture, and specific implementation of the executable code.

3.1 The Fast Multipole in Three Dimensions - Adaptive Scheme

The computational cost of the N body problem is tremendously reduced by recursively arranging nearby source elements into smaller boxes and using multipole expansion coefficients $\vec{M}_n^m$ to approximate the far field components of the velocity and vortex stretching vectors[62]. Further reduction is achieved by clustering target elements and using local expansion coefficients $\vec{L}_n^m$ to approximate the contributions of all distant sources in the computational domain.

Simple expressions for the velocity and vortex stretching vectors in term of multipole and local expansion coefficients are derived in chapter2. The hybrid adaptive error-controlled fast multipole solver introduced in this chapter uses these expressions to approximate the far-field vectors.

To facilitate the exposition of the proposed error-controlled adaptive solver, we introduce the following notation:

- $\mathcal{N}_{n_D}(I_l)$: For a given target box I_l at level l, a source box J_l at the same level belong to the neighborhood of I_l, $\mathcal{N}_{n_D}(I_l)$, if and only if the distance along the x, y, and z coordinates between the centers of the two boxes is less than or equal to $n_D W_{I_l}$ where n_D is a positive integer and W_{I_l} is the width of box I_l. For $n_D = 1$, $\mathcal{N}_{n_D}(I_l)$ consists of the blue boxes in Figure 3.1. For $n_D = 2$, $\mathcal{N}_{n_D}(I_l)$ consists of the blue and green boxes in Figure 3.1.

- $\mathcal{T}_{n_D}(I_l)$: For a given target box I_l at level l, the Interaction List $\mathcal{T}_{n_D}(I_l)$, is the set of all boxes which are children of neighbors of I_l's parent and are not neighbors of I_l (green boxes in Figure 3.2). For $n_D = 1$, the interaction list is similar to that in Cheng et al. (1999) and in the hybrid scheme of Yokota for an optimum MAC value of $\frac{1}{2}$.

- $\mathcal{H}_{n_D}(I_l)$: For a given target box I_l at level l, the Inherited List $\mathcal{H}_{n_D}(I_l)$, is the set of all boxes that belong to the interaction list of I_l ancestor, and for which the adaptive solver chooses not to perform cell-cell interaction. As the name suggest, this list is transmitted from a parent box to its children.

- $\mathcal{P}(I_l)$: Parent of box I_l.

- $N(I_l)$: Number of elements in box I_l. When referring to targets, we use $N_T(I_l)$. When referring to sources, we use $N_S(J_l)$.

- $isLeaf(I_l)$: True if I_l is a leaf box, false otherwise.

- $M(J_l)$: For a given source box J_l, $M(J_l)$ is the set of multipole expansion coefficients.

- $L(I_l)$: For a given target box I_l, $L(I_l)$ is the set of local expansion coefficients.

We note that in our case the target elements are the same as the source elements. The subscripts T and S are used to make the discussion more tractable.

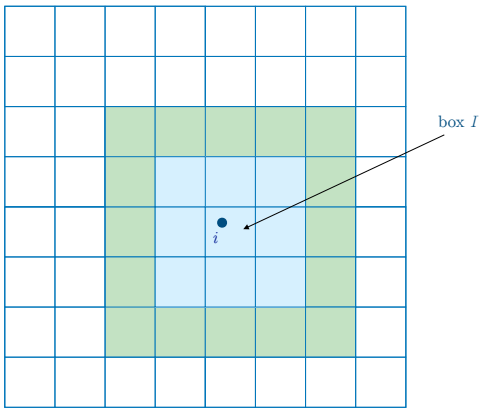

Figure 3.1: Neighborhood $\mathcal{N}(I_l)$ of box I_l containing element i, for $n_D = 1$ (blue) and $n_D = 2$. (green)

The proposed adaptive FMM scheme consists of the following steps:

1. *Construct the root box*
   Construct the smallest cubic box that contains all the vortex elements in the computational domain. This box is called the root box.

2. *Build the tree*
   Construct a hierarchy of boxes by recursively dividing the root box uniformly into smaller boxes. Refinement level 0 is equivalent to the root box. Refinement level $l+1$ is obtained by subdivision of each box at level l into eight equally sized children. The division process stops according to criterion 1 described in Table 3.1, where n_T^* and n_F^* are defined in section 3.2 and d_σ^* is defined in section 3.3.

3. *Initial expansion*
   At the finest level, all sources are expanded at their box centers to compute the far-field multipole expansion coefficients $\vec{M}_n^m$ using Eq. (2.14).

4. *Upward pass*
   Starting from the leaf boxes (boxes with no children), the multipole expansion coefficients for each source box are recursively translated from the center of the box to the center of the parent box.

5. *Downward pass*
   At the center of each box at tree levels $l \geq 2$, the local expansion coefficients are recursively calculated. Local expansion coefficients for all boxes at level 1 are first set to zero. For any target box I_l at level $l \geq 2$, initial local expansion coefficients are obtained by translating local coefficients from the center of $\mathcal{P}(I_l)$ to the center of box I_l. For any source box J_l that belongs to the interaction list $\mathcal{T}_{n_D}(I_l)$, (green boxes in Figure 3.2), We use criterion 2 to decide whether to include J_l is in the Inherited list of I_l or convert the multipole expansion coefficients at the center of J_l into local expansion coefficients at the center of box I_l using Eq. (2.19) and add them to the initial local coefficients. The downward pass stops according to criterion 3.

6. *Compute the velocity and vortex stretching vectors*
   For every target element i in the computational domain, identify leaf box I_L that contains this element.

   - Calculate the far-field velocity and vortex stretching vectors at the target position using I_L local expansion coefficients.
   - Calculate the velocity and vortex stretching vectors at the target position due to all sources contained within I_L neighbours and Inherited lists using the recursive method of the $\mathcal{O}(N \log N)$ scheme:
     - For each box $J \in \mathcal{N}_{n_D}(I_L)$ or $\mathcal{H}_{n_D}(I_L)$, the velocity and vortex stretching vectors induced at element i by all source elements contained in J are calculated by either direct summation or approximated using multipole expansion coefficients according to criterion 4.

Table 3.1: Criteria.

criterion	conditions	action
	tree building	
1	if $d_{I_l} = d_\sigma^*$ or $N(I_l) < n^*$; $n^* = \min\{n_T^*, n_F^*\}$	stop division of box I_l
	downward pass	
2	if $J_l \in \mathcal{T}_{n_D}(I_l)$ and $N_T(I_l) > \frac{n_F^{*2}}{8n_T^*}$ and $N_T(I_l) \times N_S(J_l) > \frac{n_F^{*2}}{64}$ else	$M(J_l) \to L(I_l)$ add J_l to $\mathcal{H}_{n_D}(I_l)$
	computing vector fields	
3	if $isLeaf(I_l)$ or $N_T(I_l) < n_F^*$	stop downward pass
4	if $N_S(J) < \frac{n_T^*}{8}$ or $isLeaf(J)$ else if $i \notin \mathcal{N}_{n_D}(J)$ else	direct summation fast summation proceed to next level

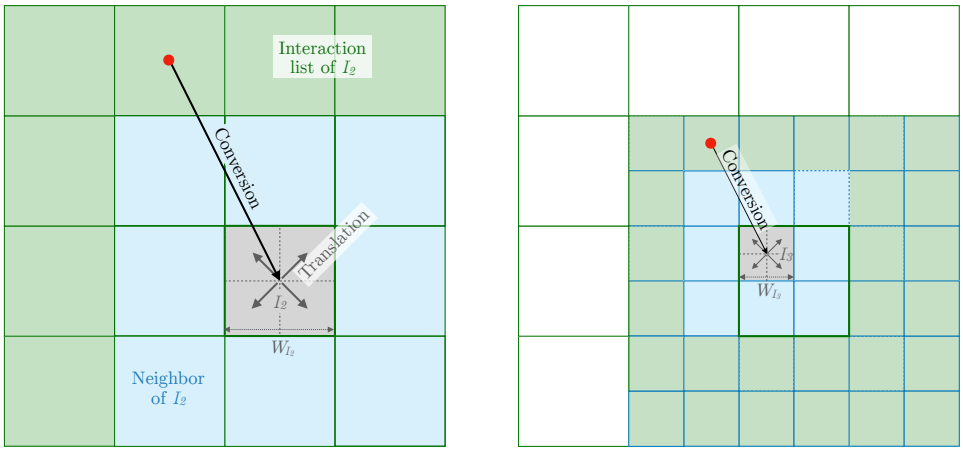

Figure 3.2: Schematic for the downward pass in step 5.

The criteria listed in Table 3.1 are designed to ensure optimal load balancing, while meeting the error threshold requirement. Proper choices of the scheme parameters, n_T^*, n_F^*, d_σ^*, in addition to the order of expansion p, are necessary for the proposed scheme to perform as intended. This is carried out through the cost and error analyses discussed next, where we account for the evolving elements distribution, in addition to other factors such as choice of the regularization function and computer architecture.

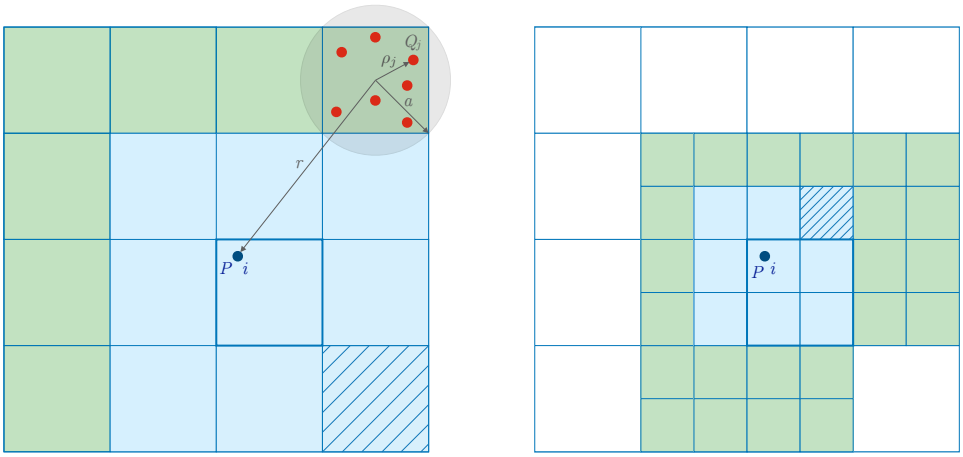

Figure 3.3: Schematic for step 6 ($\mathcal{O}(N \log N)$ FM scheme).

3.2 Introducing n_T^* and n_F^* Through Cost Analysis

The speed of the adaptive scheme is function of p, n_D, and d_m, where d_m is the tree depth. Commonly, the accuracy of the approximation can be improved by increasing p and/or n_D, at the expense of a reduction in speed due to the associated additional computational cost. For a given p and n_D, the speed of the adaptive scheme increases as d_m increases up to an optimal depth, d_m^*, above which it would be faster and more accurate to perform direct summation than to further divide the tree.

For a given p and n_D, let us consider a target box I_l at level l. Next we present a criterion, based on the number of elements contained in I_l, that allows us to decide whether or not it is faster and more accurate to further divide the box.

We denote n_T as the critical number of elements in box I_l such that for $N(I_l) \leq n_T$, the interaction between I_l and all sources in neighboring boxes, i.e. $J_l \in \mathcal{N}_{n_D}(I_l)$, is evaluated at level l using direct summation, and for $N(I_l) > n_T$, box I_l is divided into eight children, I_{l+1}, and the interactions, due to the same sources, are evaluated at level $l+1$ using both direct summation (for all sources in $\mathcal{N}_{n_D}(I_{l+1})$) and multipole expansion (otherwise). We denote n_F as the critical number of elements in box I_l such that for $N(I_l) \leq n_F$, the interaction between I_l and all sources in neighboring boxes, i.e. $J_l \in \mathcal{N}_{n_D}(I_l)$, is evaluated at level l using direct summation, and for $N(I_l) > n_F$, box I_l is divided into eight children, I_{l+1}, and the interactions, due to the same sources, are evaluated at level $l+1$ using using both direct summation (for all sources within $\mathcal{N}_{n_D}(I_{l+1})$) and local expansion (otherwise). We denote n_T^* and n_F^* as the optimal values of n_T and n_F respectively. That is, for $N(I_l) \leq n_T^*$, dividing this box into eight children and evaluating their influence using multipole expansion coefficients rather than direct summation will lower the speed of the adaptive fast solver. Similarly, for

$N(I_l) \leq n_F^*$, dividing this box into eight children and evaluating their influence using local expansion coefficients rather than direct summation will lower the speed of the adaptive fast solver. as described in section 3.1, n_T^* and n_F^* are used by the adaptive scheme to optimize the speed of the fast solver.

To facilitate the discussions that follow, we introduce the cost parameters a, b, c and d. Let a be the number of operations needed to calculate the direct interaction at a target i due to point source j, b the number of operations needed to calculate at a target i the interaction due to the sources located inside box J using multipole expansion coefficients, c the number of operations needed to convert multipole expansion coefficients, calculated at the center of a source box J, into local expansion coefficients at the center of a distant target box I, and d the number of operations needed to translate local expansion coefficients of a target box to one of its children.

3.2.1 Evaluating n_T^* Through Cost Analysis

The number of operations needed to calculate the direct interaction at target $i \in I_l$ due to all sources in neighboring boxes, i.e. $j \in J_l \in \mathcal{N}_{n_D}(I_l)$, is equal to $8\left(n_D + \frac{1}{2}\right)^3 na$, since there are $8\left(n_D + \frac{1}{2}\right)^3$ neighboring boxes containing $n = \overline{N}_S(J_l)$ sources each.

Now, we estimate, upon division of box I_l into 8 children, the number of operations required to compute the solution at target i due to the same sources, i.e. $j \in J_l \in \mathcal{N}_{n_D}(I_l)$. Noting that i belongs to one of the children, I_{l+1} at level $l+1$, the number of operations required to compute the solution at target i is the sum of direct interactions with sources in $\mathcal{N}_{n_D}(I_{l+1})$ and fast multipole interaction with sources in $\mathcal{T}_{n_D}(I_{l+1})$. The number of operations needed to calculate these interactions are respectively $\left(n_D + \frac{1}{2}\right)^3 na$ and $7 \times 8\left(n_D + \frac{1}{2}\right)^3 b$, owing to the fact that there are $8\left(n_D + \frac{1}{2}\right)^3$ neighboring boxes, each containing $\frac{n}{8}$ sources, and $7 \times 8\left(n_D + \frac{1}{2}\right)^3$ boxes which are children of box I_l but are not neighbors of box I_{l+1}.

The critical number n_T^* is then equal to the number of elements in box I_l that yields the same number of operations in the two scenarios considered,

$$8\left(n_D + \frac{1}{2}\right)^3 n_T^* a = \left(n_D + \frac{1}{2}\right)^3 n_T^* a + 7\left(n_D + \frac{1}{2}\right)^3 \times 8 \times b$$

Leading to a value of n_T^* that is independent of n_D

$$n_T^* = \frac{8b}{a} \tag{3.1}$$

We conclude that if $N_T(I_l) > n_T^*$, it is more cost effective to further divide box I_l to level $l+1$ and calculate the velocity and vortex stretching vectors induced by all sources $j \in \mathcal{N}_{n_D}(I_l)$ by direct interactions with sources in the children's neighborhoods

and using multipole expansions with sources in the children's interaction lists. Noting that b is of the order of p^2, we may express

$$n_T = c_{T2}\, p^2 + c_{T1}\, p + c_{T0} \tag{3.2}$$

where c_{T0}, c_{T1}, and c_{T2} are functions of the computer system, language, implementation, core function, etc. They also depend on what vector fields are evaluated.

The critical number n_T^* is used to minimize the number of operations needed to calculate the interactions between a given target $i \in I_l$ and $\mathcal{N}_{n_D}(I_l)$. However, for non-neighboring cells, a different criteria is used by the adaptive solver to optimize the speed of the fast summation. This criteria, which is based on the number of elements in the source box, is used by the adaptive solver to choose between cell-element and element-element interaction. If we consider a box J_l that contains $N_S(J_l)$ sources, the number of operations needed to calculate the direct interaction between J and any target $i \notin \mathcal{N}_{n_D}(J_l)$ is equal to $N_S(J_l)a$. However, since $i \notin \mathcal{N}_{n_D}(J_l)$, the interaction between J_l and i can alternatively be performed using one Multipole expansion. In this case, it is more cost effective to calculate the velocity and vortex stretching vectors using multipole expansions rather than direct summation if and only if $N_S(J_l) > \frac{b}{a} = \frac{n_T^*}{8}$.

3.2.2 Evaluating n_F^* Through Cost Analysis

The number of operations needed to calculate the direct interaction at target $i \in I_l$ due to all sources in neighboring boxes, i.e. $j \in J_l \in \mathcal{N}_{n_D}(I_l)$, is equal to $8\left(n_D + \frac{1}{2}\right)^3 na$, since there are $8\left(n_D + \frac{1}{2}\right)^3$ neighboring boxes containing $n = \overline{N}_S(J_l)$ sources each. If the number of targets in I_l is $N_T(I_l) = n$, then the number of operations needed to calculate the direct interaction between all targets $i \in I_l$ and all sources $j \in \mathcal{N}_{n_D}(I_l)$ is equal to $8\left(n_D + \frac{1}{2}\right)^3 n^2 a$.

Now, we estimate, upon division of box I_l into 8 children, the number of operations required to compute the solution at all targets $i \in I_l$ due to the same sources, i.e. $j \in J_l \in \mathcal{N}_{n_D}(I_l)$. The number of operations required to compute the solution at all target $i \in I_{l+1}$ is the sum of direction interactions with sources in $\mathcal{N}_{n_D}(I_{l+1})$ and the conversion and translation operations. The number of operations needed to calculate the direct interactions is $\left(n_D + \frac{1}{2}\right)^3 n^2 a$, owing to the fact that there are $8\left(n_D + \frac{1}{2}\right)^3$ neighboring boxes, each containing $\frac{n}{8}$ sources. The number of operations needed to perform the conversion operations is equal to $8 \times 7 \times 8\left(n_D + \frac{1}{2}\right)^3 c$ since for each of the 8 children, I_{l+1}, there are $7 \times 8\left(n_D + \frac{1}{2}\right)^3$ boxes in $\mathcal{T}_{n_D}(I_{l+1})$. In addition, the number of operations needed to perform all the translation operations is equal to $8d$.

The critical number n_F^* can then be estimated by balancing the number of operations in the above two cases

$$8\left(n_D + \frac{1}{2}\right)^3 (n_F^*)^2\, a = \left(n_D + \frac{1}{2}\right)^3 (n_F^*)^2\, a + 8 \times 7 \times 8\left(n_D + \frac{1}{2}\right)^3 c + 8d$$

leading to

$$n_F^{*^2} = 64\left(\frac{c}{a} + \frac{1}{n_I}\frac{d}{a}\right) \tag{3.3}$$

where $n_I = 56\left(n_D + \frac{1}{2}\right)^3$ is the maximum number of boxes in the interaction list. We conclude that it is more cost effective to further divide box I_l to level $l+1$ and then calculate the velocity and vortex stretching vectors using cell-cell interaction if and only if $N(I_l) > n_F^*$.

Noting the dependence of c and d on p^4, we may expressed

$$n_F^{*^2} = c_{F4}\,p^4 + c_{F3}\,p^3 + c_{F2}\,p^2 + c_{F1}\,p + c_{F0} \tag{3.4}$$

where $c_{F0}, c_{F1}, c_{F2}, c_{F3}$ and c_{F4} are functions of the computer system, language, implementation, etc. They also depend on what vector fields are evaluated.

The critical number n_F^* is used to minimize the number of operations needed to calculate the interactions between a given cell and all its neighbours. However, for non-neighboring cells, a different criteria is used by the adaptive solver to optimize the speed of the fast summation. This criteria is based on the number of elements contained in both interacting non-neighboring cells and is used by the adaptive solver to choose between cell-cell, cell-element, and element-element interactions. If we consider a source box J_l that contains $N_S(J_l)$ elements, the number of operations needed to calculate the direct interaction between J_l and any target box I_l outside $\mathcal{N}_{n_D}(J_l)$ containing $N_T(I_l)$ elements is equal to $N_S(J_l) \times N_T(I_l) \times a$. However, since I_l is outside $\mathcal{N}_{n_D}(J_l)$, the interaction between J_l and I_l can be performed using cell-cell interaction with c operations, and can also be performed using cell-element interactions with $N_T(I_l) \times b$ operations. In this case, it is more cost effective to calculate the contributions of all sources $j \in J_l$ at all targets $i \in I_l, I_l \notin \mathcal{N}_{n_D}(J_l)$ using cell-cell interation rather than direct summation or cell-element interaction if and only if $N_S(J_l) \times N_T(I_l) > \frac{c}{a} \simeq \frac{n_F^{*^2}}{64}$ and $N_T(I_l) > \frac{c}{b} \simeq \frac{n_F^{*^2}}{8n_T}$.

For a given architecture, order of expansion p, and regularization core function, the execution times a, b, c, d are measured. The values of n_T^* and n_F^* are then computed using Eqs. (3.1) and (3.3). This procedure is repeated for different values of p, and the coefficients $c_{Ti}, i = 0, 1, 2$ and $c_{Fj}, j = 0, 1, \ldots, 4$ are determined using curve fitting.

3.3 Error Analysis of the Vector Potential and Velocity Vector for the Adaptive Scheme with Regularized Vortices

In the regularized vortex method, where the vorticity field is approximated using Eq. (2.9), approximating the far-field vector potential using Eqs. (2.17) and (2.18) results in an error made up of three components:

$$\|\vec{E}_{T,\vec{\psi}}\| = \|\vec{E}_{\sigma,\vec{\psi}} + \vec{E}_{M,\vec{\psi}} + \vec{E}_{L,\vec{\psi}}\| \tag{3.5}$$

where $\vec{E}_{M,\vec{\psi}}$, $\vec{E}_{L,\vec{\psi}}$, and $\vec{E}_{\sigma,\vec{\psi}}$ are respectively the error components due to (i) truncating the multipole expansions at some order p, (ii) truncating the local expansions at some order p, and (iii) approximating the Biot-Savart kernel as a $\frac{1}{r}$ kernel. The multipole expansions truncation component, $\vec{E}_{M,\vec{\psi}}$, arises from representing the kernel $\frac{1}{r}$ at large distance using a series of multipole expansion up to order p. The local expansions truncation component, $\vec{E}_{L,\vec{\psi}}$, arises from representing the kernel $\frac{1}{r}$ at large distance using a series of local expansion up to order p and decreases exponentially as p increases. Both error components, $\vec{E}_{M,\vec{\psi}}$ and $\vec{E}_{L,\vec{\psi}}$, decrease exponentially as p increases and will eventually converge to zero as $p \to \infty$. Increasing p, however, increases the computational cost thus reducing the solver speed. We also note that the two truncation errors depend on the Multipole Acceptance Criteria, referred to as n_D. Increasing n_D improves the accuracy of the approximations at the expense of a huge reduction in speed arising from the additional cost of the associated direction summation. The error component, $\vec{E}_{\sigma,\vec{\psi}}$, arises from the fact that G_σ deviates from $\frac{1}{4\pi r}$. This error mainly depends on the size of leaf boxes (at the terminal level of the tree). In order to keep $\vec{E}_{\sigma,\vec{\psi}}$ small, the size of the leaf boxes must be larger than a minimum multiple of the core function smoothing radius. In three dimensional flows, this adds considerably to the computational cost of the FM schemes due to the associated increase in element-element interactions.

To estimate upper bounds of these error components, we consider a target element i in a leaf box I_L. The adaptive scheme can perform cell-cell, cell-element, and element-element interactions. We denote $\mathcal{M}$ as the set of all boxes J at different levels that interact with element i via cell-element interaction (Multipole Expansion) where the vector potential induced at i by all vortices contained in J is calculated using Eq. (2.17). We also denote $\mathcal{L}$ as the set of all boxes at different levels that interact with cell I_l via cell-cell interaction (Local Expansion) where the vector potential induced at i by all vortices contained in J is calculated using Eq. (2.18). The upper bounds of the error components associated with computing the vector potential (derivations are presented in Appendix A) are

$$E_{\sigma,\vec{\psi}} \leq \Gamma_{\mathcal{ML}} \beta_\sigma(r_m) \tag{3.6}$$

$$\|\vec{E}_{M,\vec{\psi}}\| \leq \frac{1}{4\pi} \frac{\Gamma_{\mathcal{M}}}{W_0} \frac{2^{d_m}}{n_D - \frac{\sqrt{3}-1}{2}} \left(\frac{\sqrt{3}}{2n_D + 1} \right)^{p+1} \tag{3.7}$$

$$\|\vec{E}_{L,\vec{\psi}}\| \leq \frac{1}{4\pi} \frac{\Gamma_{\mathcal{L}}}{W_0} \frac{2^{d_m}}{n_D + 1 - \sqrt{3}} \left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D + 1) - 1} \right)^{p+1} \tag{3.8}$$

where $\beta(r) = |G(r) - \frac{1}{4\pi r}|$, $\beta_\sigma(r) = \frac{1}{\sigma}\beta\left(\frac{r}{\sigma}\right)$, r_m is the distance that maximizes the kernel $\beta_\sigma(r)$, W_0 is the width of the root box, and $\mathcal{ML} = \mathcal{M} \cup \mathcal{L}$. Furthermore, for any set of boxes $\mathcal{S}$, we have:

$$\Gamma_{\mathcal{S}} = \sum_{j \in \mathcal{S}} \|\vec{\alpha}_j\| \tag{3.9}$$

The velocity vector is also expressed as a truncation of spherical harmonics. However it has different expressions than those obtained for vector potential, and, as such, has different error bounds. The error in the velocity vector can also be decomposed into three components, namely, the multipole truncation error $E_{M,\vec{u}}$, the local truncation error $E_{L,\vec{u}}$ and the regularizing error $E_{\sigma,\vec{u}}$. The total error can be expressed as:

$$\|\vec{E}_{T,\vec{u}}\| = \|\vec{E}_{\sigma,\vec{u}} + \vec{E}_{M,\vec{u}} + \vec{E}_{L,\vec{u}}\| \tag{3.10}$$

The upper bounds of these error components (derivations are presented in Appendix A) are

$$\|\vec{E}_{\sigma,\vec{u}}\| \leq \Gamma_{\mathcal{ML}} \kappa_\sigma(r_m) \tag{3.11}$$

$$\|\vec{E}_{M,\vec{u}}\| \leq \frac{1}{4\pi}\frac{\Gamma_{\mathcal{M}}}{W_0^2}\frac{4^{d_m}}{\left(n_D - \frac{\sqrt{3}-1}{2}\right)^2}\left(\frac{\sqrt{3}}{2n_D+1}\right)^{p+1}\left[p+2-\frac{\sqrt{3}}{2n_D+1}(p+1)\right] \tag{3.12}$$

$$\|\vec{E}_{L,\vec{u}}\| \leq \frac{1}{4\pi}\frac{\Gamma_{\mathcal{L}}}{W_0^2}\frac{4^{d_m}}{\left(n_D+1-\sqrt{3}\right)^2}\left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D+1)-1}\right)^{p+1}\left[p+2-\left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D+1)-1}\right)p\right] \tag{3.13}$$

where $\kappa(r) = \frac{1}{4\pi r^2}|1 - 4\pi q(r)|$ and $\kappa_\sigma(r) = \frac{1}{\sigma^2}\kappa(\frac{r}{\sigma})$.

For a given value of n_D, the truncation errors, $E_{M,\vec{u}}$ and $E_{L,\vec{u}}$, mainly depend on the order of expansion p. For a given required accuracy, there exists an optimal value of the order of expansion, p^*, below which the truncation errors would be higher than the maximum allowable error. Once calculated, p^* is used as the truncation order of both multipole and local expansions and to evaluate the two optimal parameters values n_T^* and n_F^* using equations (3.2) and (3.4) respectively. These optimal parameters values are used by the adaptive solver to limit the depth of the tree in order to balance near-field and far-field evaluations. For a singular (point vortices) representation of the vorticity field where the regularization error vanishes, the division process stops whenever the number of elements in a given box is less than $n^* = \min\{n_T^*, n_F^*\}$. In this case, n^* is the only parameter that controls the depth of the tree structure, i.e. $d_m^*(n^*)$. The use of a regularization core function introduces an additional error $E_{\sigma,\vec{u}}$ that mainly depends on the maximum level reached upon tree division. In fact, for a given level of accuracy, there exists a critical tree depth d_σ^* above which the regularization error exceeds the maximum allowable error. Thus, in order to maintain the desired accuracy of regularized vortex methods, the tree division must stop whenever the tree level reaches $d^* = \min(d_\sigma^*, d_m^*(n^*))$. This additional constraint would impact the FMM balance. If $d_\sigma^* > d_m^*(n^*)$, the fast solver will manage to balance the loads. On the other hand, if $d_\sigma^* < d_m^*(n^*)$, the tree structure would be too shallow and the near field evaluations dominate. In this case, the adaptive solver behaves exactly the same

as Greengard FMM[15] since the division process is halted at a given level. However, the adaptive solver will always give the best performance for the desired accuracy and used core function.

Based on the above cost and error discussions, we propose the algorithm presented in figure 3.4 to obtain the best performance for a given accuracy constraint while using a regularized core function. For a given vorticity field representation (elements distribution and core function) and a total error E_T specified by the user, the adaptive solver first determines the minimum number of samples, N^*, required to estimate the average L_2 velocity error for a 95% confidence level, and within 5% confidence interval. A sample, of size N^*, is then used to calculate the value of the optimal order of expansion p^* below which the Multipole error exceeds the total error, and the critical level d_σ^* above which the regularization error exceeds the total error. p^* is then used to calculate the values of the optimal parameters n_T^* and n_F^* according to equations (3.2) and (3.4). Once calculated, these optimal parameters values are used in the adaptive fast solver to balance the far and near-field evaluations, as described in section 3.1. As the flow-field evolves in times, N^*, p^*, n_T^*, n_F^*, and d_σ^* are re-evaluated every T times steps or whenever the calculated error takes a value greater than the pre-defined total error E_T.

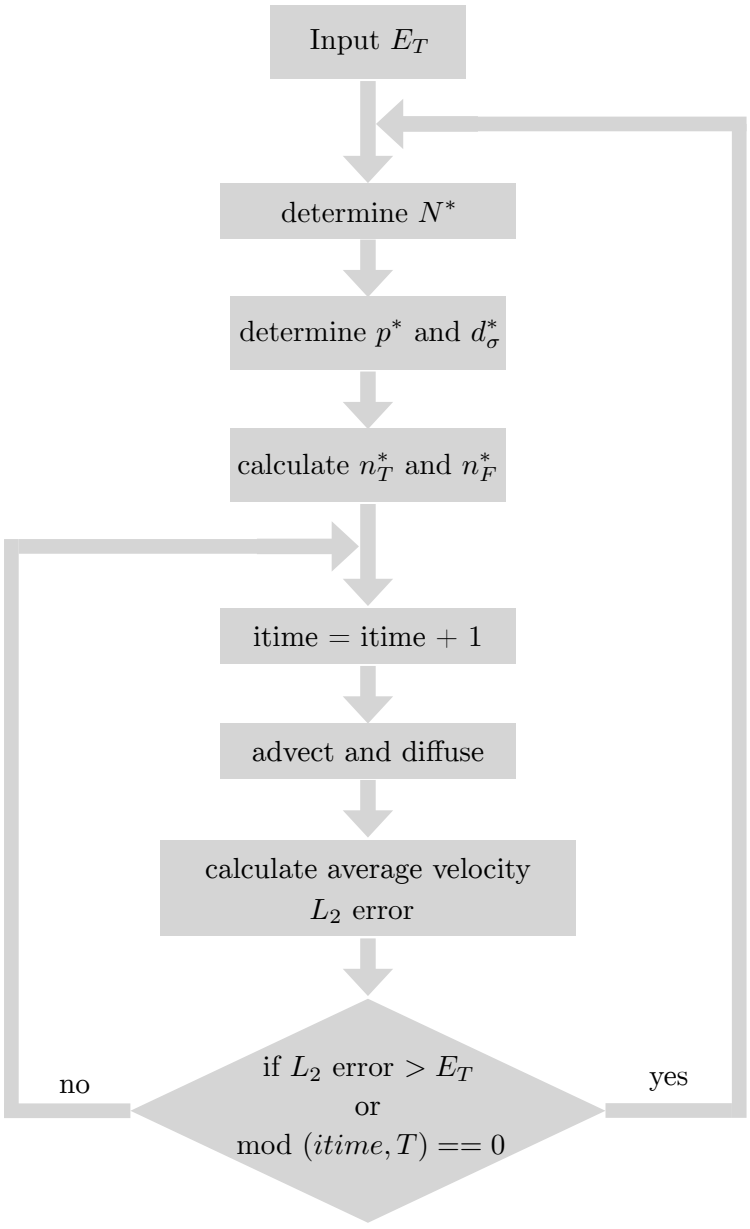

Figure 3.4: Flowchart of the error-controlled adaptive scheme.

Chapter 4

Handling the Diffusion Term in the Vorticity Transport Equation

Vortex methods[9], are grid-free Lagrangian computational methods in which the flow field is represented by a set of Lagrangian particles. They were originally developed to simulate inviscid incompressible flows at high Reynolds number[4]. However, a number of numerical diffusion model to handle diffusion effects in vortex methods have been successfully implemented. Chorin[10] introduced the Random Vortex Method (RVM) in which particles undergo a random displacement to simulate the effects of diffusion. Leonard[2] expanded the core of each vortex according to the diffusion equation. Mas-Gallic[11] introduced the particle strength exchange (PSE) and approximates the diffusion operator by an integral operator. Shankar[12] introduced the Redistribution Scheme (RSM) and accounts for the diffusion by exchanging a conserved quantity between arbitrary computational points. Furthermore, Lakkis and Ghoniem[13]introduced the smooth redistribution method (SRM) that differs from the original redistribution method[12] in assuming a smooth core function rather than the singular Dirac-delta function for the vorticity distribution. Later, they combined the SRM with the core spreading method to propose a spatially adaptive SRM[40].
Although PSE and RSM schemes have been developed since more than two decades, and despite the similarity between them (they both assign fraction of strength to neighbors particle), there is no comparison one to one between the two schemes. A comparative study is still needed in order to identify the 'best' viscous scheme. In this chapter we will discuss the simulation of diffusion using an extension to three dimensions of the smoothed redistribution method. We will also present the basic formulation of the particle strength exchange method.

4.1 Vorticity Diffusion in an Unbounded Three-Dimensional Flow Using the Redistribution Method

The diffusion scheme used throughout this paper is an extension to three dimensions of the smoothed redistribution method[19, 40, 13], based on Shankar's redistribution scheme[12] for point vortices. In the framework of operator splitting of the vorticity equation, diffusion of vorticity is carried out as a separate step. To this end, the redistribution scheme diffuses the vorticity field as a superposition of the diffusion of all the vortex elements[19], taking advantage of the linearity of the diffusion equation. The vorticity of each vortex element i is diffused by transferring a fraction f_{ij} of its strength $\vec{\alpha}_i$ to neighboring elements j that fall within a pre-determined distance $R_S h_d$, where $h_d = \sqrt{\nu \Delta t}$ is the diffusion length during a time step Δt and $\sqrt{4} \leq R_S \leq \sqrt{20}$ is a normalized search radius. In order to accurately model the diffusion process, these redistribution fractions are obtained by solving a system of linear equations governing the fractions f_{ij} such that the moments of the approximate and exact solution of the diffusion equation are equal up to a certain order. The number of equations is determined by the number of moments we choose to conserve and the stability of this method is ensured by enforcing the positivity of the redistribution fractions. Upon diffusing from time t to time $t + \Delta t$, the strength $\vec{\alpha}_j$ of each vortex element j is then calculated as the sum of the received fraction from all its neighbors i,

$$\vec{\alpha}_j \left(t + \Delta t\right) = \sum_{i \in \mathcal{N}_j} f_{ij} \vec{\alpha}_i \left(t\right) \tag{4.1}$$

where the set of neighbors $\mathcal{N}_j$ consists of all elements within a distance of $R_S h_d$ from element j. The vorticity field is reconstructed by the superposition

$$\vec{\omega}_\sigma \left(\vec{x}, t + \Delta t\right) = \sum_{j=1}^{N(t+\Delta t)} \vec{\alpha}_j \left(t + \Delta t\right) \zeta_\sigma \left(\vec{x} - \vec{x}_j\right) \tag{4.2}$$

where N is the number of elements.

Let i be a computational element characterized by vorticity strength $\vec{\alpha}_i$ and position $\vec{x}_i$. We would like to diffuse, from time t to $t + \Delta t$, the vorticity field $\vec{\omega}_i(\vec{x}, t) = \vec{\alpha}_i \left(t\right) \zeta_\sigma \left(\vec{x} - \vec{x}_i\right)$, associated with element i by transferring fractions f_{ij} of the element's strength $\vec{\alpha}_i(t)$ to M neighboring elements $j \in \mathcal{N}_i$. The fractions f_{ij} are determined such that the approximated vorticity field $\sum_j f_{ij} \vec{\alpha}_i(t) \zeta_\sigma(\vec{x} - \vec{x}_j)$ and the exact solution at time $t + \Delta t$, $\int G(\vec{x} - \vec{x}', \Delta t) \vec{\omega}_i(\vec{x}', t) d\vec{x}'$ have the same moments, expressed in a coordinate system centered as $\vec{x}_i$ as

$$\begin{aligned} & \int_{-\infty}^{\infty} x^m y^n z^p \sum_j f_{ij} \vec{\alpha}_i(t) \zeta_\sigma(\vec{x} - \vec{x}_j) d\vec{x} \\ = & \int_{-\infty}^{\infty} x^m y^n z^p \left(\int_{-\infty}^{\infty} G(\vec{x} - \vec{x}', \Delta t) \vec{\alpha}_i \left(t\right) \zeta_\sigma \left(\vec{x}'\right) d\vec{x}' \right) d\vec{x}, \end{aligned} \tag{4.3}$$

up to order $k \geq 0$, where m, n and p are non-negative integers satisfying $0 \leq m+n+p < k$, and G is the Green's function of the diffusion equation.

For a core function of order $l = k$, after removing $\vec{\alpha}_i(t)$ from both sides, equation (4.3) reduces to:

$$\sum_j f_{ij} x_j^m y_j^n z_j^p = 0 \text{ for odd } m \text{ or } n \text{ or } p \tag{4.4}$$
$$= \frac{(4\nu\Delta t)^{\frac{m+n+p}{2}} \ \gamma\left(\frac{m+1}{2}\right) \ \gamma\left(\frac{n+1}{2}\right) \ \gamma\left(\frac{p+1}{2}\right)}{\sqrt{\pi}^3} \text{ otherwise}$$

where γ is the complete gamma function $\gamma(z) \equiv \int_0^\infty e^{-t} t^{z-1}\, dt$, especially $\gamma(1/2) = \sqrt{\pi}$.

Defining the vector $\vec{r}_{ij} = \vec{x}_j - \vec{x}_i$, and shifting coordinates to the global system of coordinates, the linear system governing f_{ij} that conserves moments up to the second order is

$$\sum_{j=1}^{M} f_{ij} = 1, \quad \sum_{j=1}^{M} f_{ij} \vec{r}_{ij} = \vec{0}, \tag{4.5}$$

$$\sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,x} \vec{r}_{ij}^{\,y} = 0, \quad \sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,x} \vec{r}_{ij}^{\,z} = 0, \quad \sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,y} \vec{r}_{ij}^{\,z} = 0, \tag{4.6}$$

$$\sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,x} \vec{r}_{ij}^{\,x} = 2\nu\Delta t, \quad \sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,y} \vec{r}_{ij}^{\,y} = 2\nu\Delta t, \quad \sum_{j=1}^{M} f_{ij} \vec{r}_{ij}^{\,z} \vec{r}_{ij}^{\,z} = 2\nu\Delta t \tag{4.7}$$

For a given order of accuracy, the redistribution equations, subject to the non-negativity constraints, are solved using the non-negative least squares (NNLS) method. If the system fails to yield non-negative fractions, new elements, selected from a set of candidates uniformly distributed on a sphere of radius $R_I h_d$, are sequentially injected until a solution is obtained, where R_I is the normalized injection radius ($R_I \simeq \sqrt{10}$). Injection of a new element is carried out in a manner that maximizes the minimum distance to the existing neighbors, which ensures some level of uniformity in the elements distribution. Introducing new elements in the diffusion process will eventually act to fill the vorticity gaps created by large strain rates which cause the particles to move away from each other which degrades the spatial resolution. This will reduce the need for remeshing since the added elements will contribute to maintaining the overlap condition. For flows with very high strain rate, one can opt to reduce the time step which in turn reduces the ratio of the advection to diffusion length scales (which scales as $O(\sqrt{\Delta t})$) leading to an improvement in the core overlap condition between neighboring particles.

4.2 Vorticity Diffusion Using the Particle Strength Exchange Method

This method was introduced by Mas-Gallic[11] in 1987. The key idea is to replace the diffusion operator by an integral one. The integral operator is discretized by quadrature, using as quadrature points the locations of the particles. Using Taylor series expansions, Mas-Gallic showed that a good approximation of the Laplacian operator is given by:

$$\Delta\vec{\omega}(\vec{x}) = \frac{2}{\sigma^2}\int (\vec{\omega}(\vec{y}) - \vec{\omega}(\vec{x})) \times \eta_\sigma (\vec{x} - \vec{y})\, d^3y \tag{4.8}$$

where $\eta_\sigma(\rho) = \frac{1}{\sigma^3}\eta\left(\frac{\rho}{\sigma}\right)$, $\eta(\rho) = -\frac{1}{\rho}\frac{d}{d\rho}\zeta(\rho)$, and $\zeta(\rho)$ is a radially symmetric regularization function of order r that satisfy the following moment property:

$$\int_0^\infty |\eta(\rho)|\rho^{3+r} d\rho < \infty \tag{4.9}$$

Eq. 4.8 provides an integral approximation of the Laplace operator. Particle approximation of the diffusion can then be defined from numerical integrations of this equation. The resulting scheme is written as:

$$\frac{d}{dt}\vec{\alpha}_p = \frac{2}{\sigma^2}\sum_q (V_p\vec{\alpha}_q - V_q\vec{\alpha}_p) \times \eta_\sigma (\vec{x}_p - \vec{x}_q) \tag{4.10}$$

To model the diffusion, PSE scheme redistribute the strengths among particles in a consistent and accurate manner following Eq 4.10. Thus, the formulation of PSE is, in principle, grid-free. However, the fact that the accuracy relies strongly on the quadrature rules used for the discretized integral means that we must always have nearly uniformly spaced particle locations. The uniformity of the particle distribution is periodically restored using remeshing.

Chapter 5

The Problem of Divergence Free Vorticity

Vortex methods are grid-free methods used to solve the Navier-Stokes equation by following the evolution of the vorticity-carrying fluid elements. In these methods, the vorticity field is initially discretized using either singular point vorticies or regularized vortex blobs. In both cases, the approximated initial vorticity field is not divergence free. Hence, the difficulty of vortex methods to maintain a divergence free vorticity field originates from the initialization process. For the initial condition, the divergence can be made small enough by resolving the initial vorticity field using a sufficiently large number of elements. However, for high Reynold number, the intense vortex stretching will tend to move elements farther apart and to amplify the divergence of the vorticity field. The issue is then to analyze to what extent the divergence of the vorticity is increased by the velocity gradient field.
The fact that the approximate vorticity field either in the particle form or in its regularized form is not divergent free is by itself a difficulty that has long been a major obstacle to vortex particle method. In principal such violation of the fundamental law of fluid mechanics may result in nonphysical topology of vorticity lines (in particular the possibility of having non closed vortex lines). Moreover such a violation may leads also to the non-conservation of the basic invariants in three dimensional inviscid flows, namely, total circulation, linear angular impulses, energy, and Helicity. In fact, since $\nabla.\vec{\omega}$ is not equal to zero, the stretching term cannot be written in a conservative form $\nabla.(\vec{\omega} : \vec{u})$ and thus contribute to the production of circulation. Henceforth there is no guaranty that the circulation will be conserved.

For long time simulations, enforcing the divergence free condition is essential especially in high inertia flows. To this end, few tools have been developed. Some of these methods rely on subtracting the gradient part of the vorticity field, other recover the vorticity field from the velocity field[4]. Yet, maintaining a divergence free vorticity field remains a standing challenge in vortex methods. In this chapter, we present few relaxation schemes introduced to force the particle vorticity field to remain nearly divergence-free at all times.

5.1 Winckelmans Method

The approximated vorticity field is not generally divergence free. In fact, for a regularized representation of the vorticity, the divergence of the vorticity field is expressed as:

$$\nabla \cdot \vec{\omega}_\sigma = \sum_p \vec{\alpha}_p . \nabla \zeta_\sigma(\vec{x} - \vec{x}_p) = \frac{1}{\sigma^4} \sum_p \frac{d\zeta(r)}{dr}|_{\frac{r}{\sigma}} (\vec{\alpha}_p . \vec{u}_{r_p}), \tag{5.1}$$

where $\vec{u}_{r_p} = \frac{\vec{x}-\vec{x}_p}{||\vec{x}-\vec{x}_p||}$. It is clear that $\nabla.\vec{\omega}_\sigma$ is generally not equal to zero and its value is inversely proportional to σ^4. As σ increases, $\nabla.\vec{\omega}_\sigma$ decreases.
Initially $\vec{\omega}_\sigma(\vec{x}, 0)$ can be set to be a good representation of the vorticity field but as time evolves, this property is not maintained. Winckelmans[56, 43] proposed to reconstruct the divergence free vorticity field by taking the curl of the calculated velocity field, $\vec{\omega}_\sigma$, expressed as:

$$\tilde{\vec{\omega}}_\sigma(\vec{x}, t) = \nabla \times \vec{u}_\sigma(\vec{x}, t) = \nabla \times (\nabla \times \psi_\sigma(\vec{x}, t)) = -\nabla^2 \psi_\sigma(\vec{x}, t) + \nabla(\nabla \cdot \psi_\sigma(\vec{x}, t)) \tag{5.2}$$

$$\begin{aligned} \tilde{\vec{\omega}}_\sigma(\vec{x}, t) &= \sum_p [\vec{\alpha}_p(t) \zeta_\sigma(\vec{x} - \vec{x}_p(t) + \nabla(\vec{\alpha}^p(t) \cdot \nabla(G_\sigma(\vec{x} - \vec{x}_p(t)))] \\ &= \sum_p \left[\zeta_\sigma(\vec{x} - \vec{x}_p) - \frac{q_\sigma(\vec{x} - \vec{x}_p)}{||\vec{x} - \vec{x}_p||^3} \right] \vec{\alpha}_p \\ &+ \left[3\frac{q_\sigma(\vec{x} - \vec{x}_p)}{||\vec{x} - \vec{x}_p||^3} - \zeta_\sigma(\vec{x} - \vec{x}_p) \right] \times \frac{(\vec{x} - \vec{x}_p) \cdot \vec{\alpha}_p}{||\vec{x} - \vec{x}_p||^2} (\vec{x} - \vec{x}_p) \end{aligned} \tag{5.3}$$

Whenever $\vec{\omega}_\sigma$ becomes a poor representation of the divergence free field, Winckelmans assign new particle strengths $\vec{\alpha}_p^{new}$ by imposing $\vec{\omega}_\sigma^{new}(\vec{x}_p) = \tilde{\vec{\omega}}_\sigma(\vec{x}_p)$. This is done by solving a system of linear equation for all p:

$$\sum_q \vec{\alpha}_p^{new} \zeta_\sigma(\vec{x}_p - \vec{x}_q) = \sum_q \left[\vec{\alpha}_q^{old}(t) \zeta_\sigma(\vec{x}_p - \vec{x}_q) + \nabla(\vec{\alpha}_q^{old} \cdot \nabla(G_\sigma(\vec{x}_p - \vec{x}_q)) \right] \tag{5.4}$$

The corrected vorticity field is always divergence free independent of the choice of the core function. However, this method is quite dissipative and does not conserve the linear invariants. In fact, we have showed (derivation is detailed in appendix C) that every time we apply this method to reconstruct a divergence free vorticity field, the total circulation and linear impulse are multiplied by a factor of $\frac{2}{3}$, so we have:

$$\vec{\Gamma}^{new} = \frac{2}{3} \sum_p \vec{\alpha}_p^{old} = \frac{2}{3} \vec{\Gamma}^{old} \tag{5.5}$$

$$\vec{I}^{new} = \frac{2}{3} \sum_p \vec{x}_p \times \vec{\alpha}_p^{old} = \frac{2}{3} \vec{I}^{old} \tag{5.6}$$

According to Newton's second law of motion, the rate of change of a system's momentum is directly proportional to the net force acting on it. Hence, this scheme produces non-physical results since it acts as an external net force that dissipates energy out of the system and slow down fluid particles.

5.2 Marshal Method

According to Marshal[18], the vorticity vector, $\vec{\omega}$, can be decomposed into the sum of a divergent vector field, $\vec{q}(\vec{x},t)$, and the gradient of a scalar, $\nabla\alpha(\vec{x},t)$, such that

$$\vec{\omega} = \vec{q} - \nabla\alpha \tag{5.7}$$

In an unbounded domain, the gradient term in the vorticty equation makes no contribution to the induced velocity. This property was noted both by Novikov (1983) and Winckelmans and Leonard (1993). In fact, it also follows from the fact that $\nabla\times\nabla\alpha = \vec{0}$. For $\nabla.\vec{\omega} = 0$, the vorticity equation is expressed as:

$$\vec{\omega} = \vec{q}(\vec{x},t) - \nabla\left[\frac{1}{4\pi}\int\frac{\vec{q}'.\vec{r}}{r^3}dv'\right] \tag{5.8}$$

If the generator field, $\vec{q}(\vec{x},t)$, is represented in terms of overlapping elements centered at the control points $x_n(t)$ $n = 1, ...N$

$$\vec{q}(x,t) = \sum_n \vec{\alpha}_n(t)\zeta_\sigma(\vec{x} - \vec{x}_n(t), \tag{5.9}$$

the vorticity field can be expressed as:

$$\vec{\omega}(x,t) = \sum_n [\vec{\alpha}(t)\zeta_\sigma(\vec{x} - \vec{x}_n(t) + \nabla(\vec{\alpha}_n(t)\cdot\nabla(G_\sigma(\vec{x} - \vec{x}_n(t)))] \tag{5.10}$$

The vorticity is divergence-free for any core function, ζ_σ. The strength vectors $\vec{\alpha}_n$ are obtained by solving the matrix

$$\vec{\omega}_p = \sum_n [\vec{\alpha}_n\zeta_\sigma(\vec{x}_p - \vec{x}_n) + \nabla(\vec{\alpha}_n\cdot\nabla(G_\sigma(\vec{x}_p - \vec{x}_n))] \tag{5.11}$$

where $\vec{w}^p$ denotes the value of the vorticity at the control points $\vec{x}_p$. Hence, Marshall and Grant assign a value resulting from a global matrix calculation that insures that a divergence free vorticity field results from the particle discretization. The method utilizes a representation for the vorticity field as a set of overlapping elements, similar to the 'vortex blob' method which converges to the prescribed vorticity field as the number of control points becomes large. Yet, there is no proof that that this method converges to exact solution. Moreover, at each time step, Marshal method requires finding the inverse of a $3N \times 3N$ square matrix, where N is the number of control points. For large N, this method become infeasible.

5.3 Divergence Filtering Method

In the context of singular particle, a relaxation schemes called "divergence filtering method" was proposed by Pedrizzetti[44]. This scheme can also be used in the context of regularized particle. In his paper Pedrizzetti introduced a linear filtering feedback procedure requiring to align the vorticity field with the reconstructed solenoidal vorticity field, $\tilde{\vec{\omega}} = \nabla\times\vec{u}_\sigma$. When the strength vectors are well aligned with the reconstructed

vorticity field, the particle representation is a good approximation of the divergence free vorticity field. As such, Pedrizzetti introduced a filtering procedure which gives the strength vectors tendency to realign with the reconstructed solenoidal vorticity field, $\tilde{\vec{\omega}}$. He defined the aligned intensity as

$$\vec{\gamma}\left(\vec{x}_p(t),t\right) = \frac{\nabla \times \vec{u}_\sigma\left(\vec{x}_p(t),t\right)}{\|\nabla \times \vec{u}_\sigma(\vec{x}_p(t),t)\|}\|\vec{\alpha}_p\| \tag{5.12}$$

and introduce an equation for the evolution of the vorton intensity

$$\frac{d(\vec{\alpha}_p)}{dt} = -\eta(\vec{\alpha}_p - \vec{\gamma}_p) \tag{5.13}$$

The parameter η is the characteristic frequency of the linear filter operation. At each time step the strength vector are updated using:

$$\vec{\alpha}_p^{new} = (1-\eta\partial t)\vec{\alpha}_p + \eta\partial t\frac{\nabla \times \vec{u}_\sigma\left(\vec{x}_p(t),t\right)}{\|\nabla \times \vec{u}_\sigma(\vec{x}_p(t),t)\|}\|\vec{\alpha}_p\| \tag{5.14}$$

The divergence filtering procedure has no clear physical meaning. In fact, even at the initial stage where the divergence can be made small enough by resolving the initial vorticity field using a sufficiently large number of elements, the calculated vorticity does no align with the reconstructed solenoidal vorticity field, $\tilde{\vec{\omega}}$.

Chapter 6

Results and Discussion

In section 6.1, the convergence of the velocity and vortex stretching vectors, expressed as a truncated series of spherical harmonics, are first assessed by presenting the dependence of the L_2 error in the velocity and vortex stretching vector fields on the order of expansion, p. The performance of the adaptive solver in terms of the dependence of cost and accuracy on the various controlling parameters is investigated in sections 6.2 and 6.3. The evolution of the optimal values of these parameters along with the associated computational savings for the case of collision of two vortex rings over a reasonable time span are presented in section 6.4. In section 6.5, the adaptive solver, along with the introduced expressions, are used to assess the overall performance of three-dimensional grid-free regularized vortex methods by simulating the collision of two vortex rings, over a long period of time, for different values of Reynolds number covering the range $500 - 2000$. In section 6.6, we present a comparative study of the PSE diffusion model and the extension to 3D of the smoothed redistribution

6.1 Convergence of the FMM Expressions for the Velocity and Vortex Stretching Vectors

The convergence of the derived simple expressions for the velocity and vortex stretching vectors, expressed as a truncated series of spherical harmonics, is assessed as p, the order of expansion, is increased from 2 to 20. Evaluations are carried for a vortex elements distribution representing a vortex ring with a dimensionless core radius $a = 0.1$. The vorticity within the core of the ring is taken as a second-order Gaussian distribution

$$\vec{\omega}(r) = \frac{\Gamma}{2\pi a^2} exp^{\frac{-r^2}{2a^2}} \vec{e}_\theta \tag{6.1}$$

where $\vec{e}_\theta$ is the unit vector along the azimuthal direction, r is measured from the core center, and Γ is the circulation around the core and it is set to unity. The vorticity field, discretized using 1377744 equi-spaced elements, is approximated by

$$\vec{\omega}_\delta(\vec{x}) = \sum_i \vec{\alpha}_i \delta\left(\vec{x} - \vec{x}_i\right), \tag{6.2}$$

where the strength of element i is $\vec{\alpha}_i = \vec{\omega}(\vec{x}_i)dv$ and dv is the grid cell volume.

L_2 errors in the velocity and the vortex stretching vectors, calculated for both $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ schemes, are plotted against the expansion order in Figs. 6.1 and 6.2, respectively. The errors, which measure the departure of the solutions predicted by the $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ schemes from those obtained by direct summation, are calculated using the following L_2 norm expressions

$$L_2(\vec{v}) = \sqrt{\sum_i ||\vec{v}_f(\vec{x}_i) - \vec{v}_d(\vec{x}_i)||^2} \tag{6.3}$$

$$L_2(\vec{S}) = \sqrt{\sum_i \left|\left|\vec{S}_f(\vec{x}_i) - \vec{S}_d(\vec{x}_i)\right|\right|^2} \tag{6.4}$$

where the subscripts d and f refer to the direct and fast multipole schemes respectively, and $\vec{S} = \vec{\omega} \cdot \nabla \vec{u}$ is the vortex stretching vector. As can be observed in Figs. 6.1 and 6.2, the solutions obtained by the fast solver converge rapidly to the direct solution as the expansion order increased, where the L_2 errors decay as C^p with C ranges between 0.41 and 0.48.

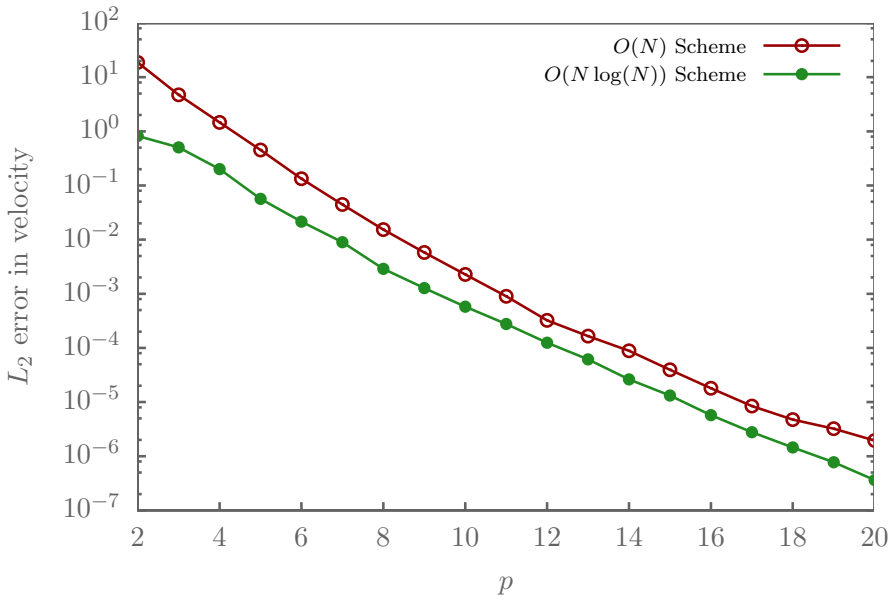

Figure 6.1: L_2 norm of the velocity error vs p for $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ schemes

6.2 Computational Cost of the Adaptive Solver

The proposed error-controlled algorithm makes use of the two optimal parameters n_T^* and n_F^* developed in section 3.2 to balance the far-field and near-field evaluations for

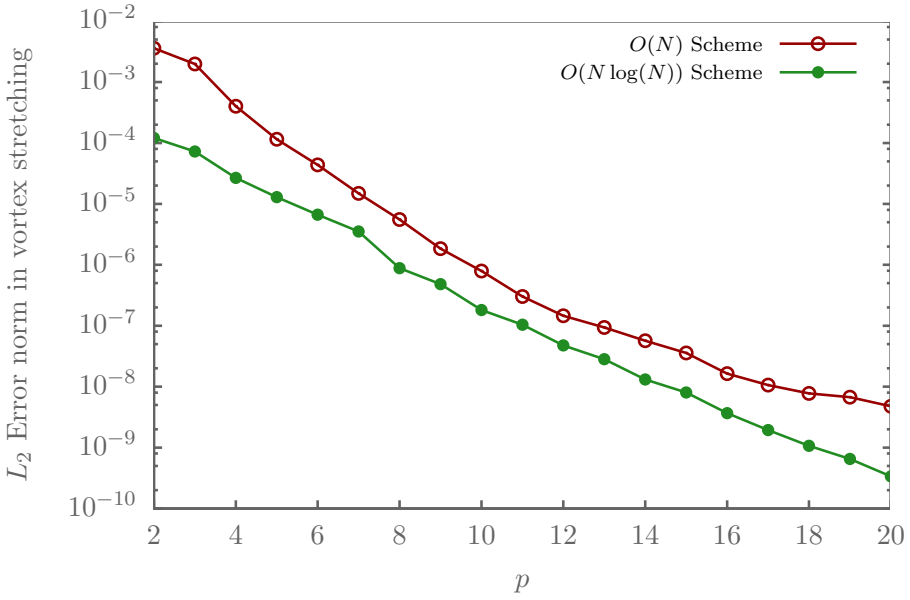

Figure 6.2: L_2 norm of the vortex stretching error vs p for $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ schemes

different architectures. For a given architecture, order of expansion p, and regularizing core function, the execution times a, b, c, d are measured. The values of n_T^* and n_F^* are then computed using Eqs. (3.1) and (3.3). This procedure is repeated for different values of p, and the coefficients $c_{Ti}, i = 0 \ldots 2$ and $c_{Fj}, j = 0 \ldots 4$ are determined using curve fitting. For example, using an Intel Xeon E5-2620 architecture (2.1 GHz, 8 physical cores) and singular vortices, the dependence of n_T^* and n_F^* on p is shown in figure 6.3 for the case of optimizing the cost of computing the velocity vector, and in figure 6.4 for the case of optimizing the cost of computing both the velocity and vortex stretching vectors. The fitted expressions for n_T^* and n_F^* are respectively $n_{T,\delta}^*(p) = 3.112p^2 + 13.35p + 19.11$, $n_{F,\delta}^*(p) = 5.897p^2 + 9.468p + 19.69$ and $n_{T,\delta}^*(p) = 4.802p^2 + 16.74p + 63.52$, $n_{F,\delta}^*(p) = 4.167p^2 + 6.689p + 13.91$, where the subscript δ is indicates the use of a singular representation of the vorticity field.

As seen in figures 6.3 and 6.4, $n_{T,\delta}^*$ and $n_{F,\delta}^*$ increase as p increases. This is due to the fact that it is more expensive to calculate cell-element and cell-cell interactions for larger p. When only the velocity vector is calculated by the fast scheme, n_F^* takes larger values than n_T^*, as shown Figure 6.3. Thus further cost reduction can be achieved by reaching higher tree level and using cell-element interaction at these levels to approximate the far-field velocity. At these levels, local expansion coefficients are not calculated since cell-cell interactions would increase the overall computational cost. However when both the velocity and vortex stretching vectors are evaluated by the fast scheme, n_F^* takes on smaller values than n_T^* for any order of expansion p, as shown Figure 6.4. Thus local expansion coefficients are calculated for all cells at leaf level and the use of cell-element interaction is the more expensive choice. In this case, the adap-

tive scheme will basically calibrate the balance between cell-cell and element-element interactions to obtain the optimum speed.

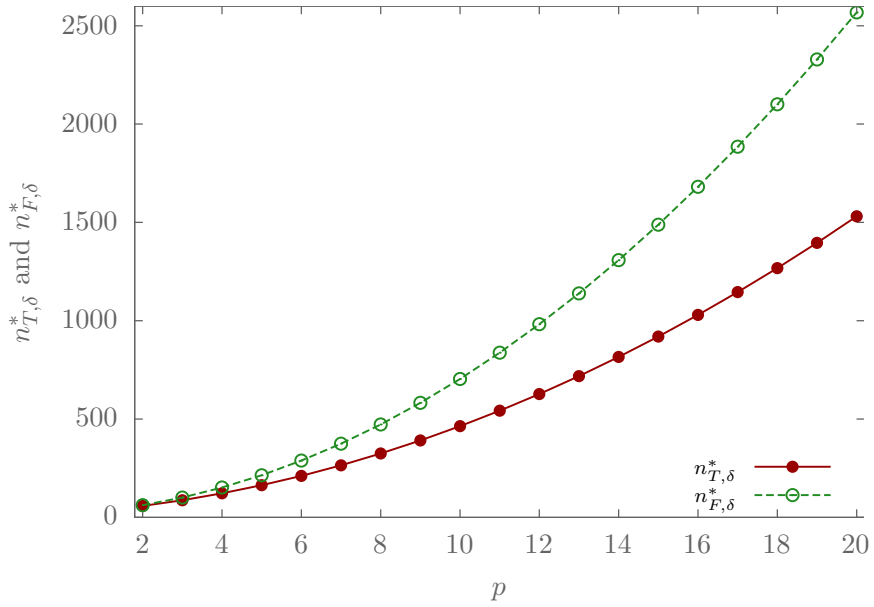

Figure 6.3: Optimum $n^*_{T,\delta}$ and $n^*_{F,\delta}$ used when calculating only the velocity vector vesus p.

The fact that $n^*_{F,\delta}$ becomes smaller than $n^*_{T,\delta}$ when both the velocity and stretching vectors are calculated can be justified as follows. Cell-cell computational cost is mainly due to conversion and translation operations which are used to evaluate local expansions coefficients at the center of each leaf cell. Once obtained, these coefficients are used to approximate the velocity, vortex stretching, and any higher order derivative of the velocity vector field. This is a fixed cost that must be incurred whether we want to calculate only the velocity vector or both the velocity and vortex stretching vectors and it is by far the most expensive component of the cell-cell evaluations. Thus calculating the vortex stretching vector, in addition to velocity vector, would not impact cell-cell computational load since the local expansion coefficients must be calculated either way. Computing the vortex stretching tensor, in addition to the velocity vector, would, however, dramatically increase the element-element and cell-element computational costs. Thus, to achieve the optimum speed, $n^*_{F,\delta}$ must decrease in order to balance element-element and cell-cell interactions. On the other hand, the corresponding decrease of $n^*_{T,\delta}$ is rather limited because both element-element and cell-element loads are increasing.

The values of n^*_T and n^*_F also depend on the regularization function used to evaluate the near-field velocity and vortex stretching tensor via element-element interaction. For a given regularization core function, $\zeta(r)$, the optimal values of n_T and n_F are related

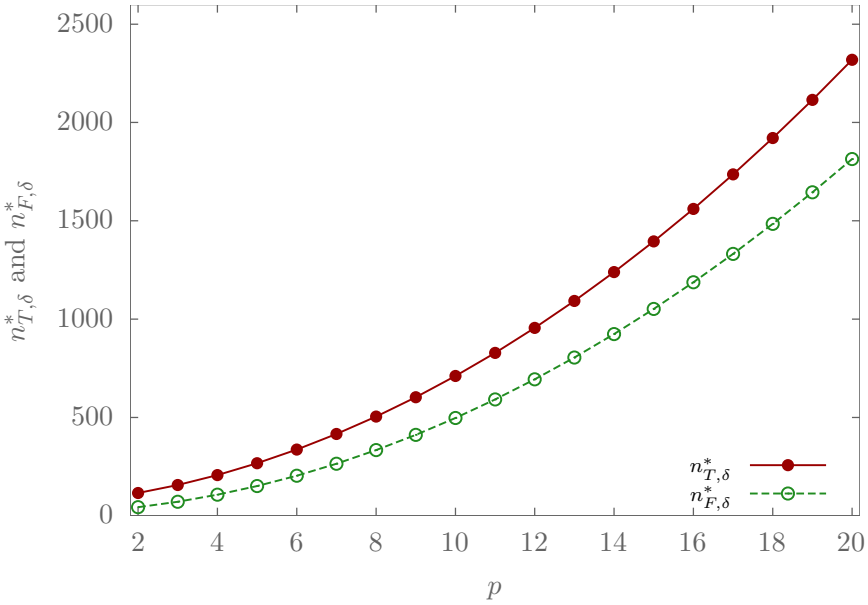

Figure 6.4: Optimum $n^*_{T,\delta}$ and $n^*_{F,\delta}$ used when calculating both velocity and stretching vectors vesus p.

to $n^*_{T,\delta}(p)$ and $n^*_{F,\delta}(p)$ as

$$n^*_{T,\sigma}(p) = \frac{n^*_{T,\delta}(p)}{R} \tag{6.5}$$

$$n^*_{F,\sigma}(p) = \frac{n^*_{F,\delta}(p)}{\sqrt{R}} \tag{6.6}$$

where $R = \frac{a_\sigma}{a_\delta}$, and a_δ and a_σ are the number of operations needed to calculate the direct interaction between two elements for the case of singular and regularized vortices respectively. Using the same architecture mentionned above, the values of R for different core functions are listed in table 6.2

Description	$4\pi\zeta(r)$	R
Dirac Function	$\delta(r)$	1.000
First order algebraic	$\frac{3}{(1+r^2)^{\frac{5}{2}}}$	1.318
Second order algebraic	$\frac{7.5}{(1+r^2)^{\frac{7}{2}}}$	1.349
Fourth order algebraic	$\frac{1}{2}\frac{5-4r^6}{(1+r^6)^{\frac{7}{2}}}$	1.38
Second order Gaussian	$\sqrt{\frac{2}{\pi}}e^{-\frac{r^2}{2}}$	1.724
Fourth order Gaussian	$\sqrt{\frac{2}{\pi}}\left(2.5-\frac{r^2}{2}\right)e^{-\frac{r^2}{2}}$	1.775
Second order Gaussian	$3e^{-r^3}$	1.532
Fourth order exponential	$1.5\left(5-3r^3\right)e^{-r^3}$	1.659

Table 1. Value of R for different core functions.

To investigate the performance of the adaptive scheme, we carried out several simulations where both velocity and vortex stretching vectors are calculated using a fourth order algebraic core function. A vortex ring with a unit radius and a core radius equal to 0.1 is considered. The initial vorticity within the core of the ring, taken as a second-order Gaussian distribution, is discretized using N singular vortex elements where N ranges between 10 thousand and 20 million. Throughout all the simulations, n_D is set to 1. This is equivalent to a typical fast multipole with a $3 \times 3 \times 3$ neighbor list. The adaptive scheme employs the criteria listed in Table 3.1 to perform cell-cell, cell-element, and element-element interactions.

The computational cost, measured in terms of CPU time on a single CPU core, is plotted versus the number of elements in Figure 6.5 for $n_F = 40, 129, 250$, and 500, using an order of expansions $p = 5$ and an optimal value of $n_T^* = 193$. Inspection of the figure reveals that, for all considered values of the number of elements, the value of n_F that optimizes the speed of the adaptive scheme by balancing near-field and far-field evaluations is $n_F = n_{F,\sigma}^*(p) = 129$ where $p = 5, R = 1.38$. For other values of n_F, the adaptive scheme suffers from load imbalance between the near-field and far-field evaluations. Element-element interactions dominate the cost for $n_F = 250$ and 500, while cell-cell interaction are dominant for the case of $n_F = 40$, which is due to the associate cost of computing local expansions for cells in a deep tree structure.

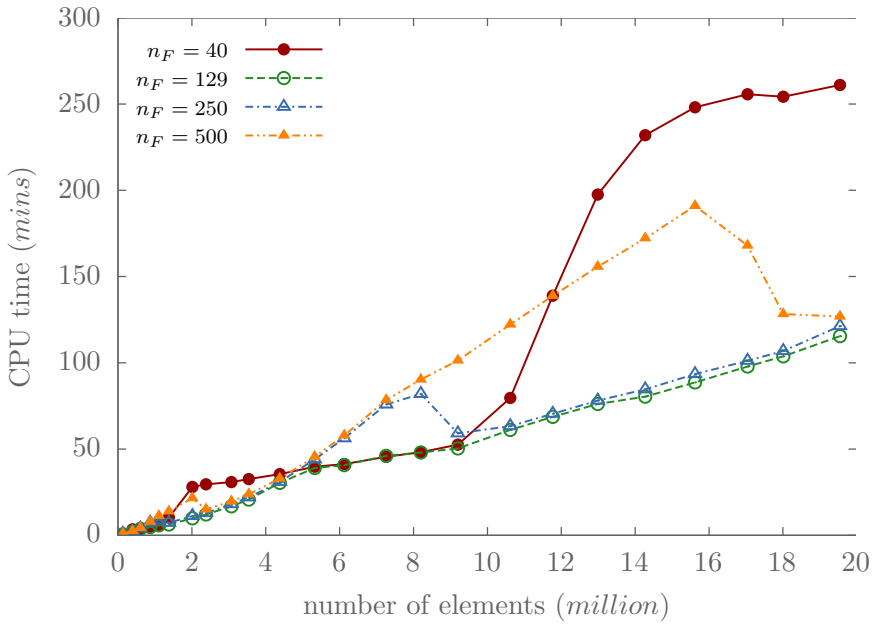

Figure 6.5: Measured simulation time versus the number of elements for different n_F using an order of expansion $p = 5$.

Figure 6.5 also shows that as n_F deviates from its optimum value, the simulation time oscillates above that of $n_{F,\delta}^* = 129$. These oscillations assume a higher peak value and

a smaller frequency as n_F further departs from its optimal value. As can be seen in the figure, the load imbalance caused a 250% increase in the CPU time for $n_F = 50$ and $N = 20$ million elements.
Figure 7.12 in Appendix D shows that similar results are obtained for $p = 8$, where the optimal value of $n_F = n^*_{F,\sigma}(p) = 285$, chosen by the adaptive solver, gives the best performance.

We conclude that the optimum case has major advantages. By limiting the depth of the tree structure and balancing near-field and far-field evaluation, the adaptive scheme optimizes the speed of the fast multipole method for a given order of expansion. This scheme is flexible and has auto-tuning capabilities on heterogeneous architectures, and can run on any machine without changing anything. Moreover, the current method will automatically choose the optimal values of the scheme parameters and save the user from this burden. In addition, the adaptive scheme scale as $\mathcal{O}(N^{1.045})$ and eliminates the number of elements dependence of the method.

6.3 Error Analysis of the Adaptive Aolver

For singular representation of the vorticity field, choosing the values of n^*_T and n^*_F according to Eqs. (3.2) and (3.4) optimizes the speed of the adaptive solver for a given order of expansion p. In this case, the accuracy of the fast summation is completely determined by the order of expansion. However, the use of a regularization core function induces an additional error, which depends on the used core function, $\zeta(r)$, and the neighborhood dimension, n_D, as discussed in section 3.3.

To investigate the accuracy of the adaptive scheme, we report the L_2 norm of the truncation and regularization errors in velocity for different values of p and choices of $\zeta(r)$, for the vortex ring considered above, where the initial vorticity distribution is discretized using $N = 1108880$ elements. For each value of p, the tree division process stops whenever the number of elements in a box become less than $n^* = \min\{n^*_T(p), n^*_F(p)\}$ without any constraint on the maximum allowable depth. In this case, the tree depth obtained upon tree division is denoted by d^*_m, which, for a given elements distribution, is a function of n^* only. Figure 6.6 shows the L_2 error norm versus p for the fourth order algebraic core function. The L_2 error norm, which measures the departure of the solution predicted by the adaptive scheme from that obtained by direct summation, is computed using the following expression

$$L_2(\vec{v}) = \sqrt{\sum_i ||\vec{v}_f(\vec{x}_i) - \vec{v}_d(\vec{x}_i)||^2} \tag{6.7}$$

where the subscripts d and f refer to the direct and fast multipole schemes respectively. Figures 7.8 and 7.9 in Appendix D show L_2 error norm plotted against p for a second order algebraic core function and second order Gaussian core function respectively. One can observe in Figures 6.6 and 7.8 and 7.9 that there exists a critical value of p below which the total error is dominated by the truncation error and above which it

is dominated by the regularization error. As can be seen in all cases, the truncation error monotonically decreases exponentially, while the regularization error (and thus the total error) decreases in jumps. As p increases, these jumps occur when the tree depth, $d_m^*(n^*)$, decreases (for example from 6 to 5 as shown in the figure) in response to an increase in n^*. In the range of p where the regularization error is dominant, increasing p beyond a certain value $p^*(d_m^*(n^*))$ (in Figure 6.6, p^* =3 for $d_m^*(n^*)$=6 and 9 for $d_m^*(n^*)$ =5) will only lead to an improvement in accuracy if also accompanied by a drop in the tree depth, $d_m^*(n^*)$. This is because the regularization error depends mainly on the core function and on the distance to neighbor cells, which is related to the tree depth $d_m^*(n^*)$, which in turn assumes staircase dependence on p.

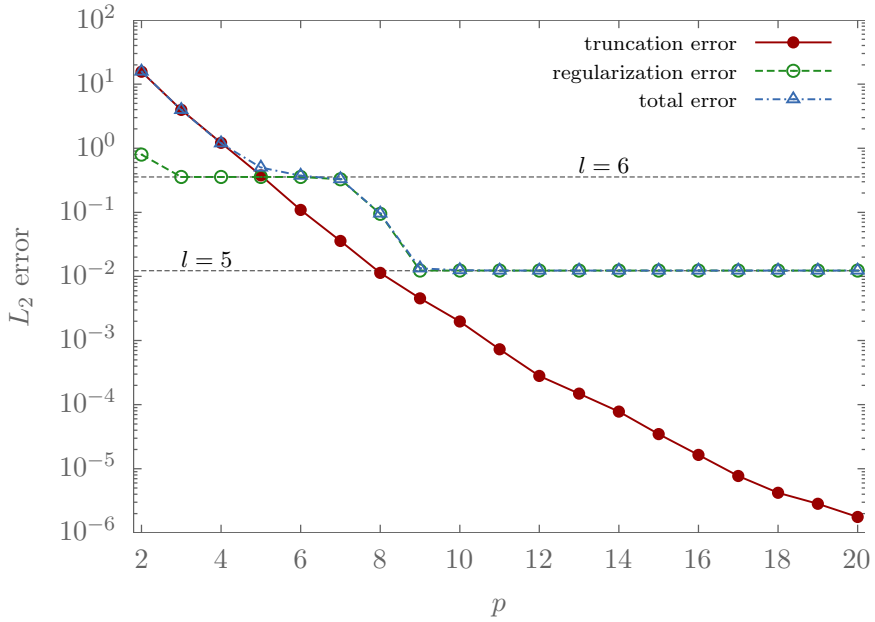

Figure 6.6: L_2 norm for the truncation and regularization error versus p using a fourth order algebraic core function in the context of adaptive scheme.

Figure 6.7 shows the L_2 norm of the truncation and regularization errors versus tree depth, d_σ, for a fixed order of expansion $p = 5$ using a fourth order algebraic core function. Figures 7.10 and 7.11 in appendix D show the variation of these errors with d_σ for a second order algebraic core function and second order Gaussian core function, respectively. It is clear that as the tree depth d_σ increases, the regularization error increases while the truncation error remains constant. The increase in the regularization error is because increasing the tree depth d_σ will decrease the distance between neighboring cells and thus increase κ_σ.

Figure 6.8 shows the regularization error versus tree depth, d_σ, for the three core functions considered above and for a fixed order of expansion $p = 5$. It can be seen that the regularization error resulting from the use of the fourth order algebraic core function is always smaller than that obtained for a second order algebraic core function. Moreover, the second order Gaussian core function is more accurate than both algebraic

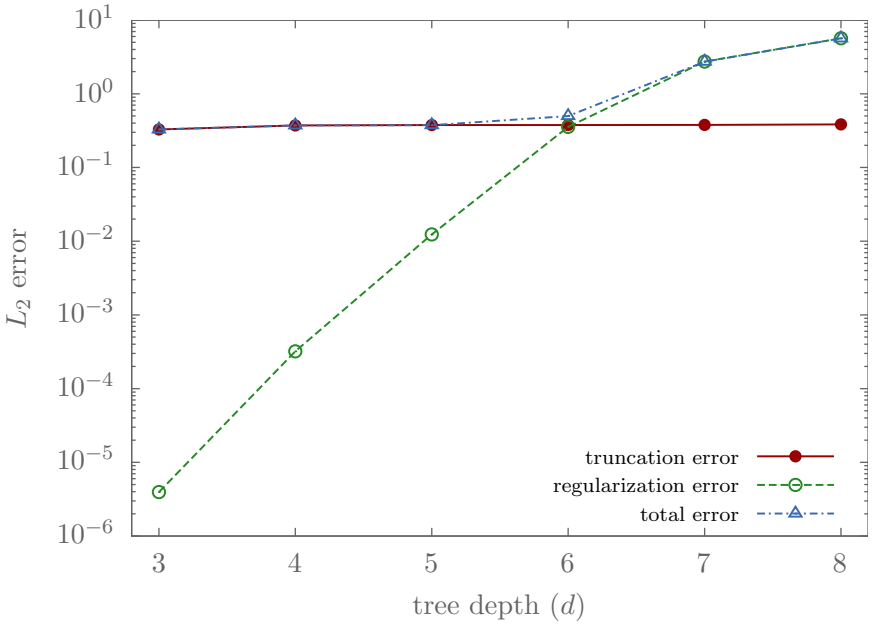

Figure 6.7: L_2 norm of the truncation and regularization errors versus d_σ for a fixed order of expansion $p = 5$ using a fourth order algebraic core function in the adaptive scheme.

functions for small d_σ. However, as d_σ increases, the second order Gaussian core function quickly loses its accuracy to become the least accurate at tree depths larger than 5. Thus, in the low accuracy regime, the second order Gaussian core function is the preferred choice among the core functions considered.

We conclude that, for a regularized representation of the vorticity field, it is inefficient to increase the order of expansion p unless it is accompanied by a drop in the tree depth since this will only increase the computational cost without improving the accuracy. The proposed hybrid scheme realizes optimal speed for a given accuracy by automatically computing optimal values of the scheme parameters as follows. The algorithm first determines the optimal tree depth, d_σ^*, beyond which the regularization error exceeds the user defined error threshold. For this optimal level, the algorithm then computes the optimal order of expansion, p^*, above which increasing p will only increase the computational cost without improving the total error. As shown in the flowchart of figure 3.4, d_σ^* and $p*$ are first computed by evaluating the errors on a representative sample of vortices. Then, n_T^* and n_F^* are evaluated as a function of p^*, using Eqs. (3.2) and (3.4).

Next, we present a case study to investigate the evolution of the adaptive solver key parameters in order to achieve the optimal efficiency in term of speed and accuracy.

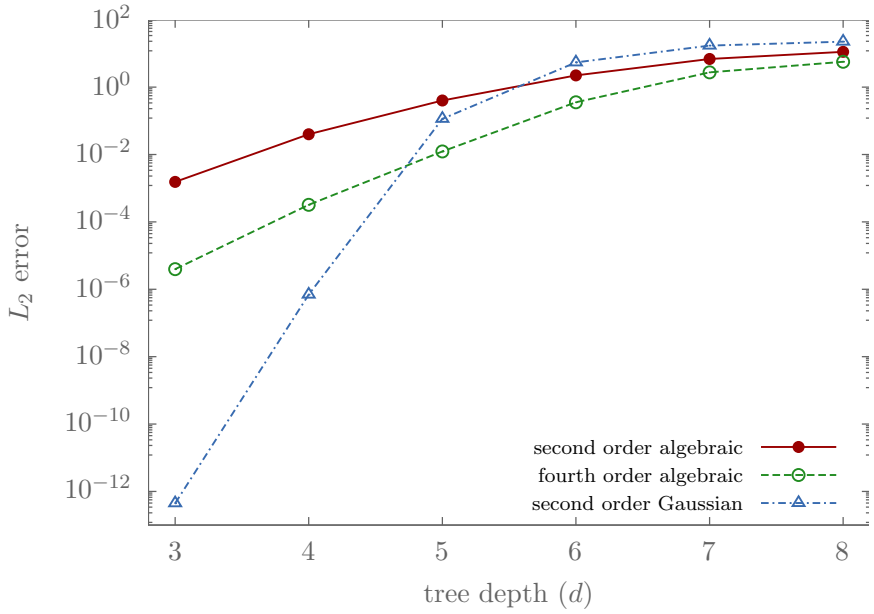

Figure 6.8: L_2 norm of the regularization error, E_σ, versus d_σ for three different core functions. The order of expansion is $p = 5$.

6.4 Case Study: Evolution of the Optimal Values of Adaptive Solver Key Parameters

In this section, we explore the performance of the proposed error controlled hybrid adaptive scheme in simulating, using a regularized grid-free vortex method, the collision of two identical axisymmetric vortex rings for Reynolds number $Re = 1000$. Initially, each of the two rings is centered on the x axis and is at an angle of 10 degree from the xy plane oriented towards the z axis, as depicted in figure 6.9. The vorticity of the rings, initially along the azimuthal direction, assumes a second-order Gaussian distribution

$$\vec{\omega}_0(r) = \frac{\Gamma}{2\pi a^2} exp^{\frac{-r^2}{2a^2}} \vec{e}_\theta \tag{6.8}$$

where r is the distance from the center, $\vec{e}_\theta$ is the unit vector along the azimuthal direction and a is the dimensionless core radius, taken to be equal to 0.1. The circulation, Γ, around the core is set to unity. The initial vorticity distribution is represented by 159400 equally spaced elements located at the centers of the cells of a uniform radial mesh constructed for each ring. The strength of each particle is set to the product of the value of the vorticity at the particle position and cell volume, $\vec{\alpha}_i = \vec{\omega}_0(\vec{x}_i)dv$. The initial vorticity field is approximated by

$$\vec{\omega}_\sigma(\vec{x}) = \sum_i \vec{\alpha}_i \zeta_\sigma \left(\vec{x} - \vec{x}_i\right), \tag{6.9}$$

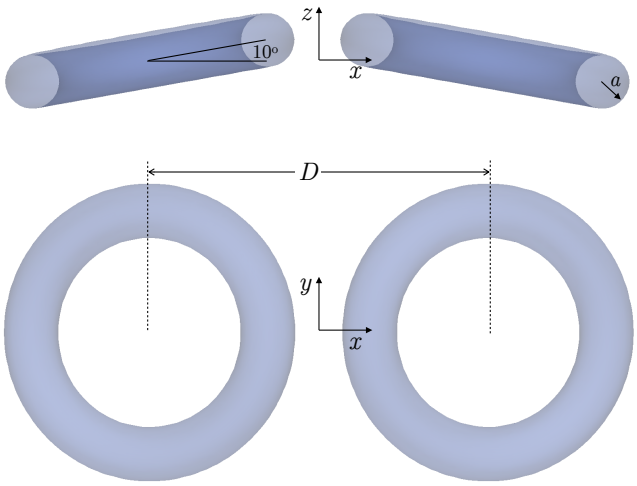

Figure 6.9: Setup of the vortex rings at initial time.

where a fourth order regularizing function with smoothing radius σ is used; $\zeta_\sigma(\rho) = \frac{1}{\sigma^3}\zeta(\frac{\rho}{\sigma})$, $\zeta(\rho) = \frac{3}{8\pi}\frac{5-4\rho^6}{(1+\rho^6)^{\frac{5}{2}}}$, and $\sigma = 2h_0$ where $h_0 = \sqrt{10\nu\Delta t}$ is the initial elements spacing.

We numerically solve this advection-diffusion problem using a (completely) grid-free vortex method that relies on operator splitting to separately handle advection and diffusion. In the advection step, the elements are convected by the local velocity field using a second order Runge-Kutta time integration scheme with a time step of $\Delta t = 0.1$ and the strength vector is strained and tilted by the local rate of strain tensor. The velocity and stretching terms are calculated using the adaptive fast multipole algorithm with an order of expansion, p^*, and tree depth , d^*_σ, calculated by the adaptive solver to meet the accuracy condition specified by the user in terms of a maximum allowable error, E_T. For the diffusion step, we use the three-dimensional Smooth Redistribution Method (SRM) presented in [67].

As presented in the flowchart of figure 3.4, for a given bounding error E_T, the adaptive solver first determines the minimum number of samples, N^*, required to estimate the average velocity error for a 95% confidence level, and within 5% confidence interval. A sample, of size N^*, is then used to calculate the optimal order of expansion p^* below which the Multipole error would take higher value than the total error, and the critical tree depth d^*_σ above which the regularization error would exceed the total error. The optimal order of expansion p^* is used to calculate the two critical numbers n^*_T and n^*_F introduced in section 3.1 and used by the adaptive solver to balance the far and near-field evaluations. This operation is repeated every 50 time steps, or whenever the calculated error takes a value greater than the pre-defined total error $E^*_T = 10^{-3}$ (see Fig 3.4).

In order to assess the efficiency of the fast solver, the CPU time and the average L_2 velocity error are measured every time step, and p^* and d^*_σ are recorded. Fig

6.10 shows, to the left, the average L_2 norm of the velocity error versus time, and to the right, the CPU times for evaluating the local expansion coefficients and direct summation versus time. At $t = 0$, the average L_2 norm of the velocity is equal to 8.84×10^{-4} which is smaller than the E_T^*. The solver automatically satisfies the error bound condition by choosing $p^* = 4$ and $d_\sigma^* = 6$. As time increases, the total error decreases since the average distance between neighboring elements increases which reduces the regularization error. Moreover, the cpu time increases as the number of elements increase. At $t = 35$, the optimal order of expansion p^* dropped from 4 to 3, which immediately resulted in more than 50% savings in the CPU time of the direct evaluation (from 1067 to 475). This is accompanied by a jump in the velocity error from 3.6×10^{-4} to 8.86×10^{-4} which is still smaller than E_T^*. For $t > 35$, the velocity error decreases and the CPU time increases due the increase of the number of elements. The adaptive solver continuously updated, and in an automated fashion, the parameters p^*, n_F^*, n_T^* and d_σ^*, to minimize the cost while meeting the accuracy condition, which resulted in considerable savings in the computational cost.

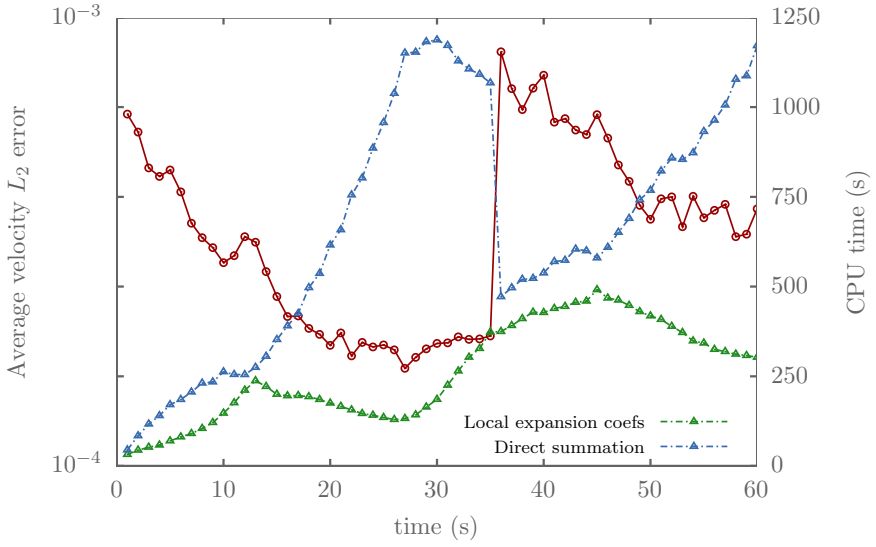

Figure 6.10: Variation with time of the average velocity L_2 error (left) and the CPU times for computing the local expansion coefficients and the direct summation (right) showing the automated adjustment of d_σ^* and p^* to reduce the computational cost while meeting the error requirement ($E_T \leq 10^{-3}$). The direct summation cost includes both the velocity and the vortex stretching computations.

6.5 Case Study: Collision of Two Identical Axisymmetric Viscous Vortex Rings

In this section, we present transient simulations, over a long period of time, of the collision of two vortex rings for different values of Reynolds number covering the range $Re = 500-2000$. We chose this problem to demonstrate the performance of the method because it has been the subject of numerous experimental and numerical studies. Kida et. al[68] numerically investigated the interaction of two identical circular viscous vortex rings using a spectral method with $64\times64\times64$ grid points. Their study explores the collision of two vortex rings at various collision angles and for initial values of Re up to 1153 over a period of time up to 60 seconds. They observed two reconnections, referred to respectively as "fusion" and "fission", the existence of which persisted over the considered ranges of viscosity and collision angle. Moreover, the collision of two vortex rings moving side by side has been studied experimentally by Kambe and Takoa [69]. They were the first to report the occurrence of the two reconnections with the second reconnection taking place only within a narrow range of the initial linear impulse. Fohl and Turner[70] also experimentally studied the collision of two rings traveling along intersecting paths in water for a $Re \simeq 4000$. They reported the existence of a critical approach angle above which fusion and fission were always observed. Their experiments showed that while fusion always occurs below this critical angle, fission does not always follow.

In this section, we present transient simulations, over a long period of time, of the collision of two vortex rings for different values of Reynolds number covering the range $Re = 500 - 2000$. The vortex method used to numerically solve this advection-diffusion problem is completely grid-free. The operator splitting scheme employs a second Runge-Kutta time integration scheme for advection and a novel extension to three-dimensions of the redistribution method[12] for the diffusion step. In the advection step, the velocity and stretching terms are calculated using an adaptive fast multipole algorithm that combines both $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$ schemes with an order of expansion p equal to 5. The code uses the expressions developed throughout this paper and is optimized in order to reduce the calculation efforts. The elements are convected by the local velocity field and the strength vector is strained and tilted by the local stretching term using a second order Runge-Kutta method with a time step of $\Delta t = 0.1$. In the diffusion step, the 3D grid-free redistribution scheme simulates the diffusion of each particle, over a time step, by transferring non-negative fractions of its strength to its neighboring vortices in a manner that satisfies the zeroth, first, and second moments of the diffusion equation. In the process, the scheme adaptively introduces new particles in gaps introduced by large strains in the advection step. The vorticity field is approximated as

$$\vec{\omega}_\sigma(\vec{x}) = \sum_i \vec{\alpha}_i \zeta_\sigma(\vec{x} - \vec{x}_i) \tag{6.10}$$

where a fourth order regularizing function with smoothing radius σ is used; $\zeta_\sigma(\rho) = \frac{1}{\sigma^3}\zeta(\frac{\rho}{\sigma})$, $\zeta(\rho) = \frac{3}{8\pi}\frac{5-4\rho^6}{(1+\rho^6)^{\frac{5}{2}}}$, and $\sigma = 2.5h_0$ where $h_0 = \sqrt{10\nu\Delta t}$ is the initial elements

spacing (and the injection radius).

Initially, the two vortex rings are set up as shown in Fig. 6.9. The rings are inclined by an angle of 10 degree towards the z axis, with their centers on the x axis. Taking the vortex ring radius R as a characteristic length scale and the circulation around its core, Γ, as a reference circulation, the dimensionless core radius of each ring and the distance separating the centers of the two rings are chosen to be $a = 0.1$ and $D = 2.8$, respectively. The reference time is $T = R^2/\Gamma$ and the reported time is made dimensionless by dividing by T. Initially the vorticity of the ring is along the azimuthal direction and depends on r, the distance from the core center, as a second-order Gaussian distribution of Eq. (6.11), where $\vec{e}_\theta$ is the unit vector along the azimuthal direction. The initial vorticity distribution is represented by particles located at the cell centers of an equispaced radial mesh constructed for each ring. The number of mesh points increases as we move outwards to maintain the radial distance between neighboring points approximately the same equal to h_0. Each particle is assigned with a strength vector $\vec{\alpha}_i$ equal to the value of the vorticity at particle position multiplied by the volume of the cell.

The results are presented in terms of iso-surfaces of vorticity norm and vorticity lines at several representative stages of the time evolution for $Re = 500$. These are shown in Figs. 6.11 - 6.12, where the levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm. For $Re = 500, 1000, 1500, 2000$, we present the following time dependent diagnostics: (i) the evolution of the total kinetic energy (Fig. 6.20), (ii) the circulation around interacting vortex tubes during various reconnection events (Figs. 6.14, 6.15, and 6.16), (iii) the evolution of the error in the divergence of the vorticity (Fig. 6.18), (iv) growth in the number of particles as a function of time (Fig. 6.21), and (v) the following flow invariants (Fig. 6.17) : total circulation, linear impulse, and angular impulse. We also show the impact of reducing the time step on the divergence of the vorticity for the case $Re = 2000$ (Fig. 6.19).
It should be noted that the degrees to which the method conserves the flow invariants and maintains the vorticity field to be divergence free are commonly used metrics to assess the convergence of the method.

6.5.0.1 First, Second and Third Reconnection of the $Re = 500$ Case

Figures 6.11-6.12 show the iso-surfaces of vorticity norm and vorticity lines projected onto the (xy) plane (as seen from z direction) for the case $Re = 500$. The figures show that as the two vortex rings move along their axes due to self-induction, they undergo rotation toward the (yz) plane due to mutual induction. In effect, the rings approach each other and eventually collide at around $t = 2$. At the collision (yz) plane, viscous diffusion causes cancellation of vortex lines carrying vorticity of opposite signs, and at the same time the outermost vortex lines, that possess the lowest vorticity, start to connect forming two symmetric "bridges" of high vorticity at the back of the rings and two symmetric "threads" at the front (Fig. 6.12, $t = 2$), where the front and back sides are defined as in Fig. 6.13. Vorticity cancellation at the interaction zone is accompanied by a strong converging flow caused by vorticity away from the zone. This flow constantly brings the innermost part of the rings closer to each other which

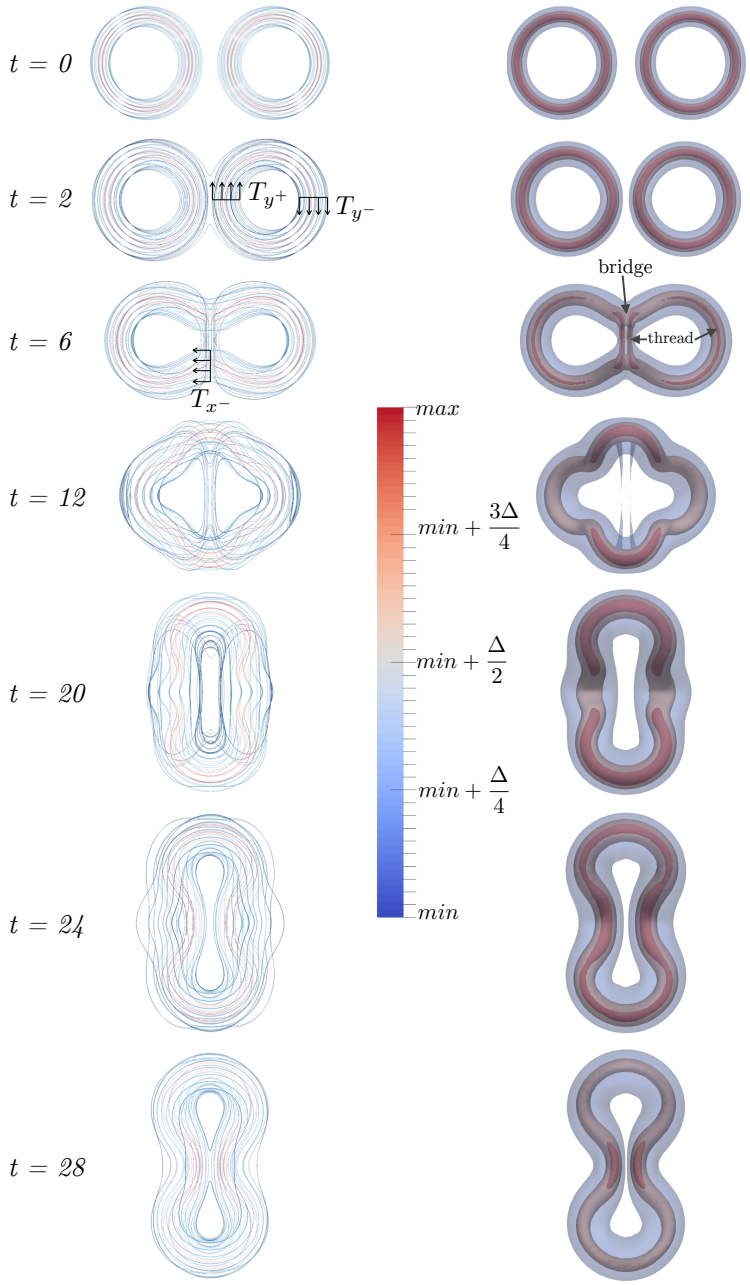

Figure 6.11: Iso-surfaces of vorticity norm and a random collection of vorticity lines at several representative stages of evolution for $Re = 500$ seen from z direction. The levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm

enhances the cancellation rate. In consequence, the number of reconnected vortex lines increases with time forming a single big distorted ring (Fig. 6.11, $t = 6$). Annihilation of vorticity in the interaction zone by viscous cross-diffusion continues to take place

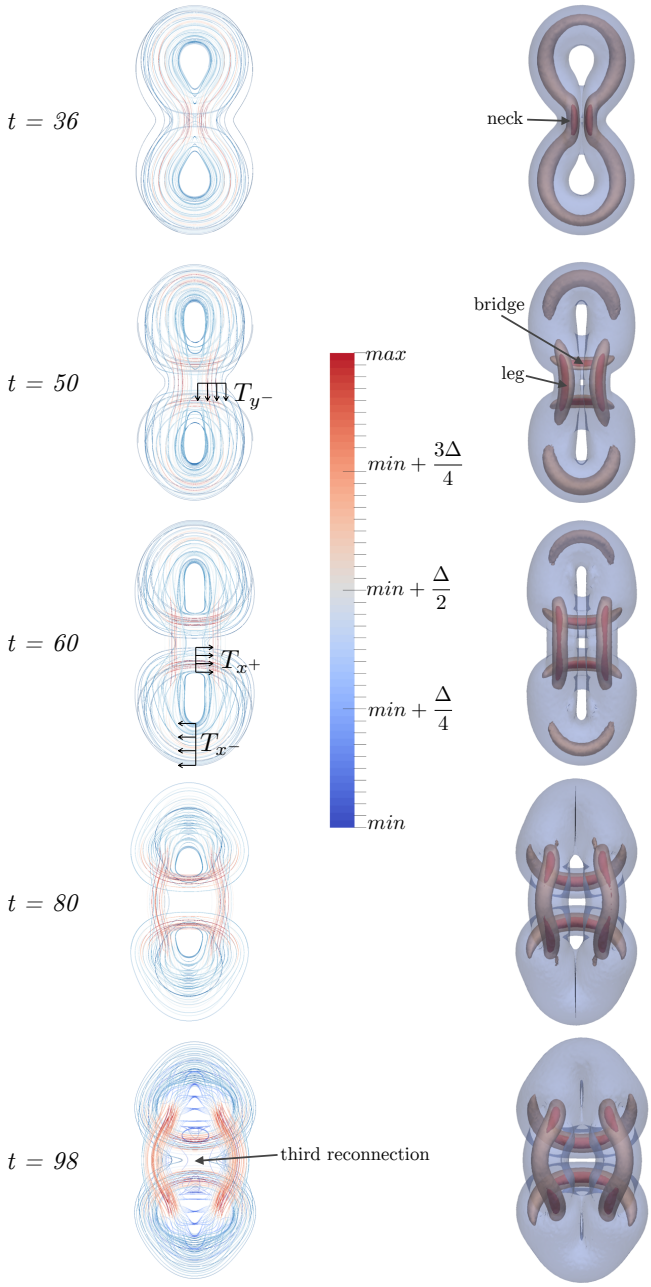

Figure 6.12: Iso -surfaces of vorticity norm and a random collection of vorticity lines at several representative stages of evolution for $Re = 500$ seen from z direction. The levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm

between the "threads" and the main distorted ring until the two "threads" disappear completely. This continues to take place until the two rings completely fuse into a single

ring at $t \sim 20$ (Fig. 6.11). This reconnection process, referred to as vortex fusion, has been observed and described in detail by Kida et al.[68] for $Re = 577$ and 1153.

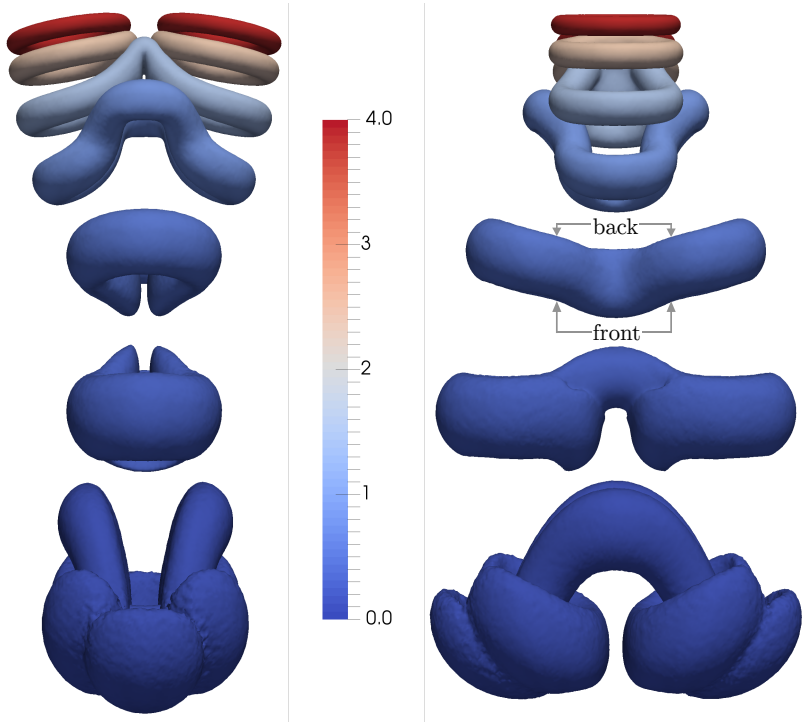

Figure 6.13: Iso -surfaces of vorticity norm at several representative stages of evolution for $Re = 500$ seen from x and y directions. The level of the iso-surface is 25% of the instantaneous maximum of the vorticity norm

At short times ($t < 10$ for the case $Re = 500$), the circulation is nearly constant along vortex tubes, as seen in Fig. 6.14. In fact, during the rings fusion process, the rate of decay of circulation $\left|T_{y^+}\right|$ is balanced by an equal circulation T_{x^-} gain. The circulations $\left|T_{y^+}\right|$ and T_{x^-} are computed respectively as vorticity integrals over the xz and yz sections shown in Fig. 6.11. This is clearly seen from Fig. 6.15 between $t = 2$ and $t = 8$ where 90% of the vortex tubes have been reconnected. These observations are in agreement with results reported by Winckelmans and Leonard[56] for $Re = 400$ and also by Kida et. al[68] for $Re = 577$ and 1153.

After the first reconnection is completed, the distorted vortex ring keeps moving in the negative z direction. Moreover, the self-induced velocity field causes stretching of the vortex tubes at the "bridges" along the y direction so that the two "bridges" rotate around the x-axis forming a plane ring that moves in the negative z direction, as shown in Fig. 6.13. Meanwhile the vorticity distribution also causes the outermost vortex tubes to approach each other, and the ring will have a "gourd-shaped form"[68] at around $t = 24$ as seen in Fig. 6.11. These tubes will continue to approach each other (narrowing of the "neck") until they contact at around $t = 28$, triggering the second reconnection.

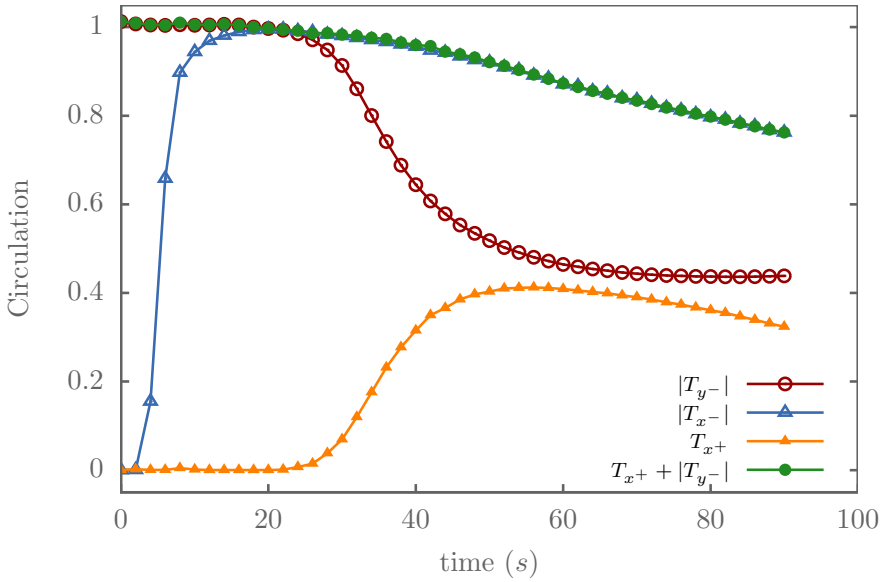

Figure 6.14: Time-development of circulation around interacting vortex tubes during the first and second reconnection for $Re = 500$

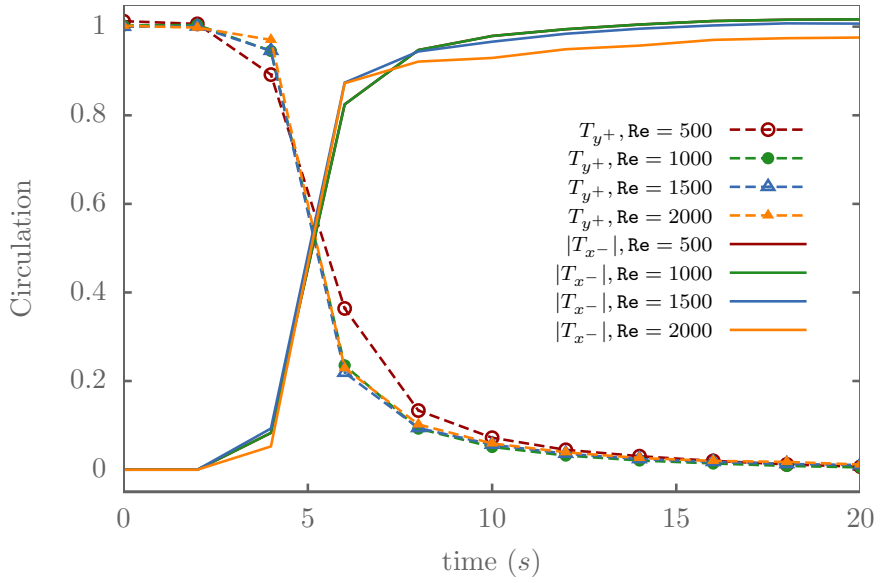

Figure 6.15: Time-development of circulation around interacting vortex tubes during the first reconnection for different values of Re

Similar to the first reconnection, the second reconnection takes place via "bridges", forming at the front of the vortex ring, that connect the two tubes of opposing vorticity direction. The second reconnection is, however, slower than the first. This is because,

in contrast with the first reconnection, the second reconnection cannot be maintained due to the absence of an external converging flow. This causes the second reconnection to stop early at $t = 54$, where two rings, connected by two "legs", emerge out of the distorted ring (Fig. 6.12, $t = 50, 60$). At this time, only 46% of the total circulation is contained in the connecting "bridges" (Fig. 6.14), while the remaining uncanceled 54% are contained in the connecting "legs". This vortex "fission" process, in which the ring seems to reverse its evolution by splitting back into two rings, has been experimentally observed in [70] and numerically predicted in[68]. The reported time and speed of these processes are different than ours due to difference in the initial conditions.

Note however that the sum of the circulation passing through the "bridges" (T_{x^+}) and the absolute value of that passing through the "legs" ($\left|T_{y^-}\right|$) is always equal to the total circulation (which in this case is T_{x^-}), as shown in Fig. 6.14. The circulations T_{x^+}, $\left|T_{y^-}\right|$, and T_{x^-} are computed respectively as vorticity integrals over the yz, xz, and yz sections shown in Fig. 6.12. Fig. 6.14 also shows that T_{x^-} decreases with time due to viscous cross diffusion. In fact, at around $t = 98$, the total circulation is decreased by 25%.

The vorticity in the growing "bridges" keeps increasing until it pushes the "legs" back to the positive z direction (Fig. 6.13). This in turn increases the "neck" curvature and the two rings formed upon the second reconnection approach each other, triggering the third reconnection that would take place at $t \simeq 98$ (Fig. 6.12, $t = 90$).

6.5.0.2 The Cases $Re = 1000$, 1500, and 2000

Prior to the first reconnection, the global motion of the vortex rings, seems to be independent of the value of Re. The first reconnection starts at $t \simeq 2$ and the cancellation of vorticity in the interaction zone is complete for all cases (see Figures 7.2 - 7.7 in Appendix A). We note, however, that the rate of vorticity cancellation is higher for larger values of Reynolds number, as seen in Fig. 6.15. This is because, despite the fact that viscosity is smaller for larger Re, the opposing vortex tubes in the interaction zone are more stretched for the larger Re case (larger vorticity), so that the net effect is a higher vorticity cancellation rate. At later times, viscosity has a bigger impact on the evolution of the vorticity field. This is manifested in a smaller approach velocity of the two rings for lower Re, resulting in a delay in the second reconnection. Fig. 6.16 shows that the second reconnection for $Re = 1000$ starts at $t \simeq 24$, which is ~ 4 earlier than that of $Re = 500$. Moreover, the second reconnection is incomplete for higher Re, with the remaining vortices having larger circulation. In fact, for $Re = 1000$, 78% of total circulation is left uncanceled compared to 54% for $Re = 500$. For $Re = 1500$ and 2000, the portion of the reconnected vortex tube is negligible and more than 98% of total circulation is left uncanceled. Moreover, for $Re = 1000$, 1500, and 2000, the vorticity field leads to stretching of the vortex tubes in the x-direction which prevents any further tube reconnection. This behavior is more prominent for larger Re where the percentage of reconnected vortex tubes decreases dramatically. This is in contrast with what we observed for $Re = 500$, where the vorticity distribution pushes back the "legs" to the positive z direction, triggering a third reconnection.

These results are also in agreement with the experiments presented by Fohl and turner[70], who reported the existence of a critical angle, equal to 16°, below which

the second reconnection does not always occur. For an angle of 10°, our simulations show that the second reconnection takes place for $Re = 500$ and 1000. However, for $Re = 1500$ and 2000, it does not. Further studies are needed to identify the parameters which completely define the occurrence of the second reconnection.

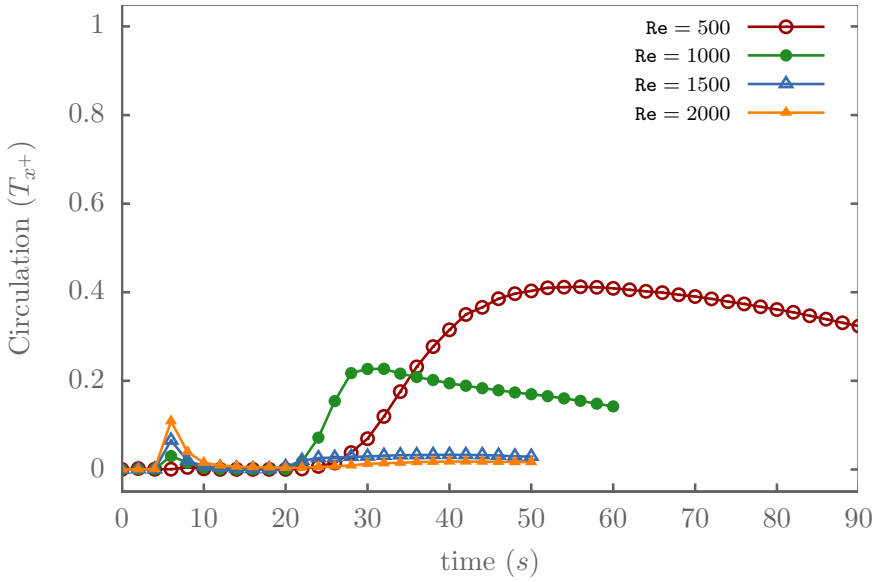

Figure 6.16: Time-development of circulation around interacting vortex tubes during the second reconnection for different values of Reynolds number

6.5.0.3 Global Flow Invariants, Vorticity Divergence, and Total Kinetic Energy

A common measure of the accuracy of vortex methods is how well they conserve the various invariants. The x, y and z components of the total vorticity are plotted against time in the first row of Fig. 6.17 for the different values of Re. Variation with time of the x and y components of the linear impulse, and the relative % error in its z component are plotted in the second row of Fig. 6.17. In addition, the x, y and z components of the angular impulse are plotted vs time in the third row of the same figure. The figures show that these quantities are conserved to within $\sim$ 98%, where the larger errors are commonly incurred for larger Re. This is attributed to the stronger inertia of the larger Re cases which results in larger strain rates that cause the particles to move away from each other which degrades the spatial resolution. Impact of inertia on the accuracy is also assessed by quantifying the departure of the divergence of the vorticity vector from zero. Fig. 6.18 shows that the normalized L_2 norm of the divergence of the vorticity decreases in time for all Reynolds numbers. As can be seen in the figure, departure from the zero vorticity divergence is larger for the case with larger Re, with the peak taking place during the fusion of the two vortex rings, where vortex stretching is the characterizing mechanism. For the initial condition, the divergence was made

small enough by resolving the initial vorticity field using a sufficiently large number of elements. As time increases, the vorticity divergence increases between $t = 0$ and $t = 2$ due to the dominance of inertia effects at early stages. In fact, although we didn't use any particular scheme to enforce the divergence free condition, a nearly divergence free vorticity field was maintained by the method. As time increases, although the vortex stretching tend to amplify the vorticity divergence, the diffusion process, modeled by the redistribution method, played a stabilizing role which damped the growth of the vorticity divergence throughout all the simulation and for all studied values of Reynolds number. In fact, the diffusion scheme will always contribute to filling the gaps by injecting new particles. Filling the gaps, as required by the redistribution scheme to model diffusion, acts against the large strain rates that tend to move elements farther apart. This effectively improves the overlap between neighboring elements and works towards maintaining the vorticity field to be divergence free. However, for large values of Re, and for moderate values of the time step (and default elements spacing), the strain rates are so large that the redistribution scheme will not be able, on its own, to fill the gaps properly and ensure overlap. Fig. 6.17 shows that this problem starts showing up at $Re = 2000$. In this case, maintaining the divergence-free vorticity field remains a challenge and the only existing cost-effective way of dealing with this challenge is re-projection. Another way, which is computationally expensive, is to reduce the time step. Fig. 6.19 shows that increasing the resolution of the method by reducing the time step led to a significant drop in the vorticity divergence during the fusion event. In fact, since the injection length is proportional to $\sqrt{\Delta t}$ and the advection length is proportional to Δt, decreasing the time step will eventually decrease the ratio of the advection length to injection length and will result also in an increase of the number of elements injected per unit time. These together will act to maintain the spatial resolution and dampen the variation of L_2 norm of $\nabla.\vec{\omega}$. Decreasing the time step is, however, expected to help in maintaining a divergence free vorticity field only up to some point. For high Reynolds number flows, where the intense vortex stretching results in complex vorticity fields, the elements injection in the redistribution scheme will be unable to maintain the desired spatial resolution. In this case, the core overlapping condition is no longer satisfied and the numerical method rapidly loses its accuracy.

Due to viscous dissipation, the total kinetic energy decreases in time, as shown in Fig. 6.20. The figure also shows that the rate of decay is larger for smaller Reynolds number because of the higher viscosity.

6.5.0.4 Computational Cost

Figure 6.21 shows that the growth in the number of elements with time obeys a nearly linear trend for all the cases studied. This increase is due to the fact that the redistribution scheme, used to simulate diffusion, injects new elements as neighbors of existing particles at the edge of the domain and/or in gaps created by large straining. The injection mechanism is carried out in a manner that promotes uniformity in the elements locations while conserving the various moments of the vorticity diffusion equation. Figure 6.21 also shows that the number of elements increases at a higher rate for larger values of Re. This is because stronger convection creates more gaps which must be filled by the redistribution scheme. Similar trends were observed by

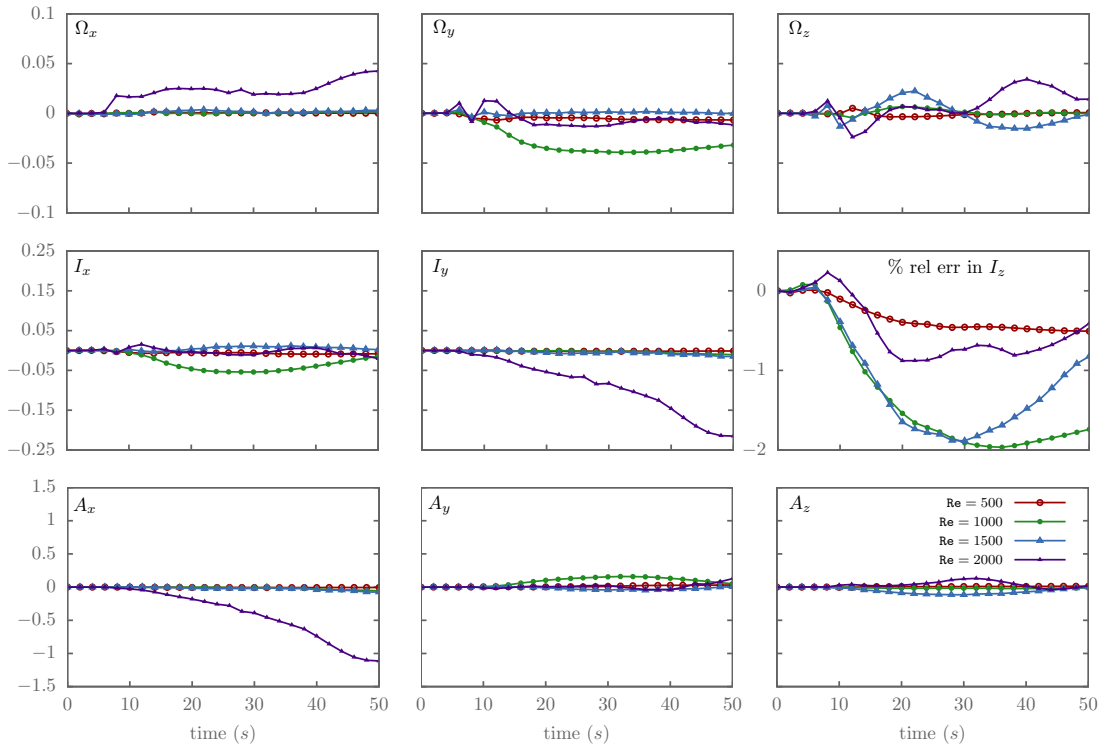

Figure 6.17: The x, y, z components $(\Omega_x, \Omega_y, \Omega_z)$ of the total vorticity vs time for different values of Re. The x, y components of the total linear impulse (I_x, I_y) and the % relative error in the $z-$component of the total linear impulse vs time for different values of Re. The x, y, z components of the total angular impulse (A_x, A_y, A_z) vs time for different values of Re.

Shankar [12] for the two dimensional flow over an impulsively started cylinder. We finally point out that optimized injection and neighbor search algorithm, in addition to parallel implementation, allows the cost of the diffusion step in terms of CPU time to be a fraction of that of the advection step.

6.6 Comparative Study of the PSE Diffusion Model and the 3D Extension of the Smoothed Redistribution Diffusion Model

In this section, we present a comparative study of PSE and the 3D extension of the smoothed redistribution (SRM) model in simulating the collision of two vortex rings for Reynolds number $Re = 500$. Initially, each of the two identical axisymmetric vortex rings is centered on the x axis and is at an angle of 10 degree from the xy plane oriented towards the z axis, as depicted in figure 6.9. The vorticity of the rings, initially along

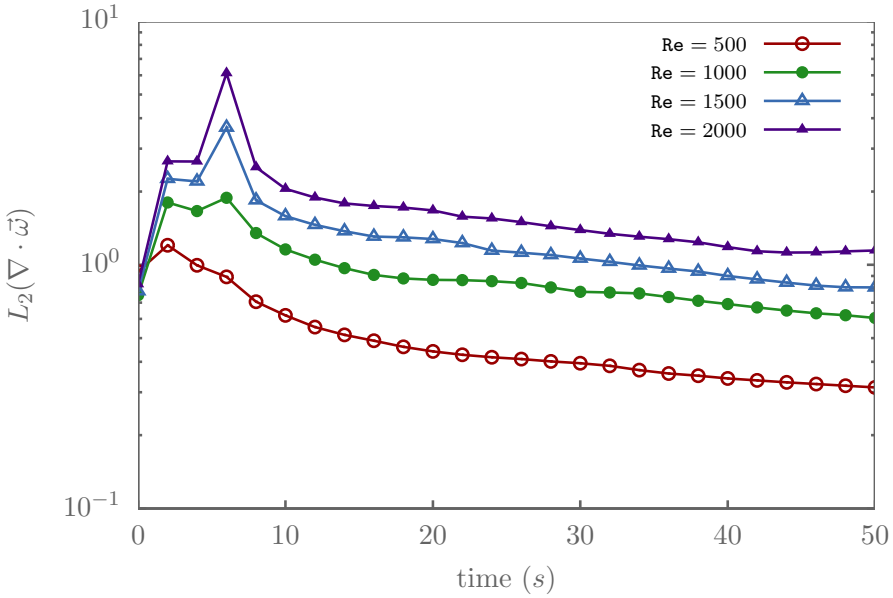

Figure 6.18: L_2 norm of the vorticity divergence vs time for different values of Reynolds number. The norm is normalized by its value at $t = 0$.

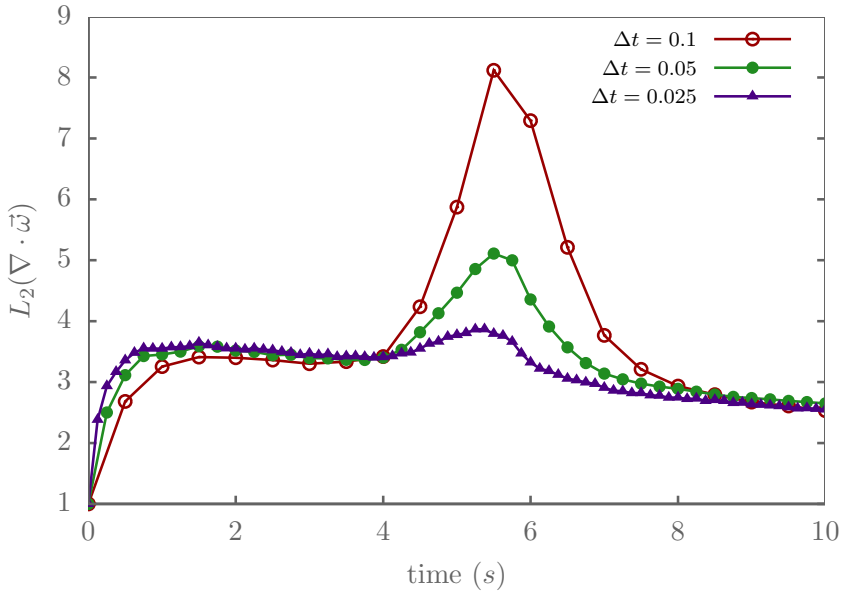

Figure 6.19: L_2 norm of the vorticity divergence vs time for different values of time step. The norm is normalized by its value at $t = 0$.

the azimuthal direction, assumes a second-order Gaussian distribution,

$$\vec{\omega}_0(r) = \frac{\Gamma}{2\pi a^2} exp^{\frac{-r^2}{2a^2}} \vec{e}_\theta \tag{6.11}$$

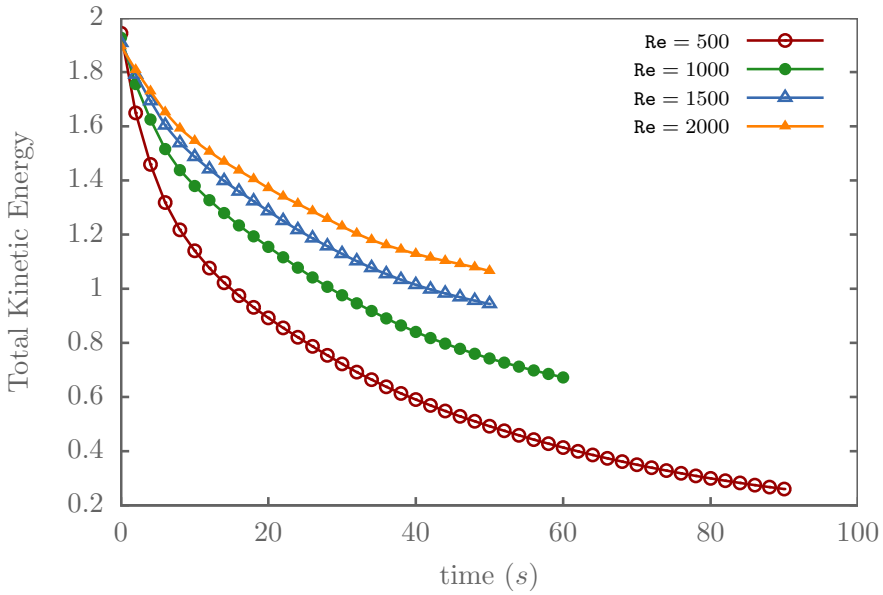

Figure 6.20: Time-development of kinetic energy of the interacting vortex rings for different values of Reynolds number

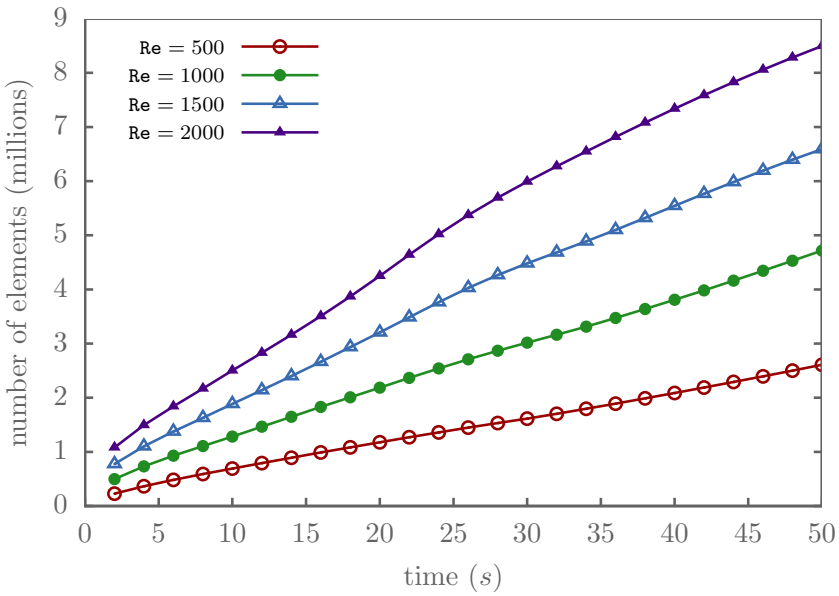

Figure 6.21: Number of elements vs time for different values of Reynolds number.

where r is the distance from the center, $\vec{e}_\theta$ is the unit vector along the azimuthal direction and a is the dimensionless core radius, taken to be equal to 0.1. The circulation, Γ, around the core of the ring is set to unity. The initial vorticity distribution is represented by 48144 equally spaced elements located at the centers of the cells of a uniform

radial mesh constructed for each ring. The strength of each particle is set to the product of the value of the vorticity at the particle position and cell volume, $\vec{\alpha}_i = \vec{\omega}_0(\vec{x}_i)dv$. The initial vorticity field is approximated by

$$\vec{\omega}_\sigma(\vec{x}) = \sum_i \vec{\alpha}_i \zeta_\sigma\left(\vec{x} - \vec{x}_i\right), \tag{6.12}$$

where a fourth order regularizing function with smoothing radius σ is used; $\zeta_\sigma(\rho) = \frac{1}{\sigma^3}\zeta(\frac{\rho}{\sigma})$, $\zeta(\rho) = \frac{3}{8\pi}\frac{5-4\rho^6}{(1+\rho^6)^{\frac{5}{2}}}$, and $\sigma = 2.5h_0$ where $h_0 = \sqrt{10\nu\Delta t}$ is the initial elements spacing.

For the PSE scheme case, M_4' remeshing scheme with $h = h_0 = 0.05$ is performed every 10 time step (with the smallest particle strength kept $\|\vec{\omega}\| = 10^{-6}$). Moreover, the cutoff distance for PSE diffusion is taken equal to 3.2σ. For the SRM scheme, we didn't use any remeshing scheme and we didn't merge nearby elements no matter how close they are. Moreover, all elements obtained upon diffusion are kept no matter how small they are. However, only elements with strength less than 10^{-6} are diffused by the SRM scheme.

The results show that similar results are obtained by the two diffusion models. In fact, both models were able to went through the "fusion" and "fission" process, and both schemes were able to identify a third vortex reconnection that took place at $t \simeq 98$. Moreover, both schemes were able to maintain a nearly divergence free vorticity field throughout the simulation. In addition, all linear invariants are nearly conserved in both simulations.
Fig 6.22 shows the number of elements versus time for both schemes. It is clear that, for the SRM case, the number of elements increased dramatically and reaches a value of 5.2 million at $t = 100$. This is due to the fact the SRM scheme will always inject new element to properly model the diffusion and we didn't apply any procedure to merge nearby elements or to redistribute elements with small strengths. On the other hand, the increase of the number of elements in the PSE scheme is very modest compared to that of the SRM scheme. In fact, the number of elements at $t = 100$ is around 1 million which is five time smaller than the number of elements obtained by the SRM scheme. This will considerably increase the computational cost of the advection step for the SRM case.
The computational cost of the advection and diffusion steps, measured in terms of CPU time on a single CPU core, is plotted versus time in Fig 6.23 for both diffusion schemes. The results show that, although the number of diffused elements in the PSE case is much smaller than that of the number of elements in the SRM case, the CPU time of the PSE diffusion step is larger than the CPU time of the PSE diffusion step. Moreover, Fig 6.23 shows that the CPU time of the diffusion step for the PSE case is even larger than the CPU time of the advection step in the PSE case. In fact, the evaluation of the diffusion term in the PSE model requires $O(N^2)$ computational operation where N is the number of elements. Although we used a small cutoff diffusion distance of 3.2σ, the CPU time of the diffusion step is approximatively twice that of the advection step for the PSE case. However, the overall CPU time (advection & diffision) of the SRM

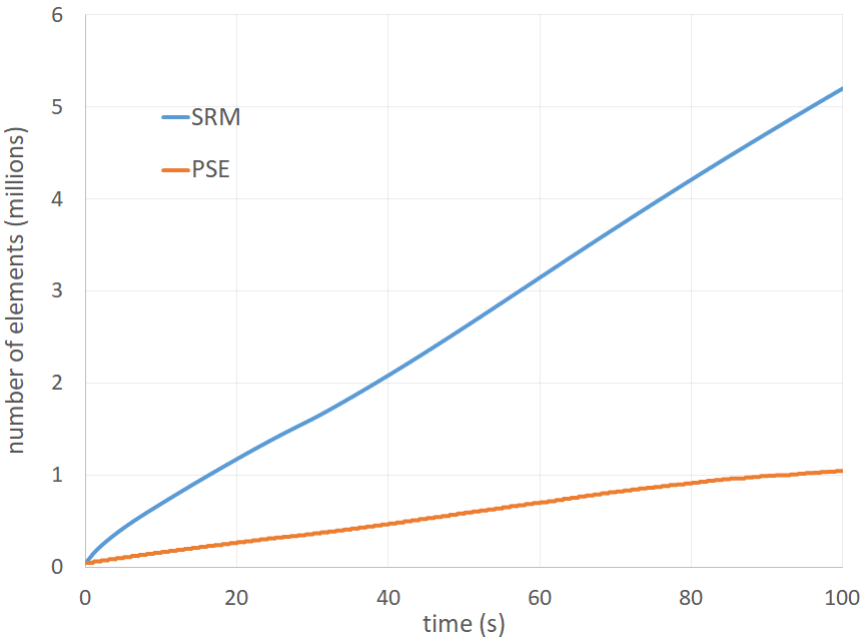

Figure 6.22: Number of elements vs time.

diffusion model is much larger than that of the PSE. This is related to the additional computational cost of advecting a very large number of elements in the SRM case.
Fig 6.24 shows the L_2 norm of the vorticity divergence versus time. As can be seen in this figure, although we didn't use any particular scheme to enforce the divergence free condition, a nearly divergence free vorticity field was maintained by both diffusion methods. Both schemes played a stabilizing role which damped the growth of the vorticity divergence throughout the simulation. However, during the fusion of the two vortex rings, where vortex stretching is the characterizing mechanism, the divergence of the vorticity in the PSE scheme case takes on higher value than that observed in the SRM case.
The relative % error in the z components of the linear impulse is plotted against time for the two diffusion models is shown in Fig 6.25. The figure shows that the linear impulse is better conserved in the SRM scheme. In fact, the relative % error is around 4.6% at $t = 100$ for the PSE case, while it is only 0.6% for the SRM case.
Fig. 6.26 shows that, throughout the simulation, the rate of vorticity cancelation and the decrease in total circulation is approximately the same for both diffusion scheme.

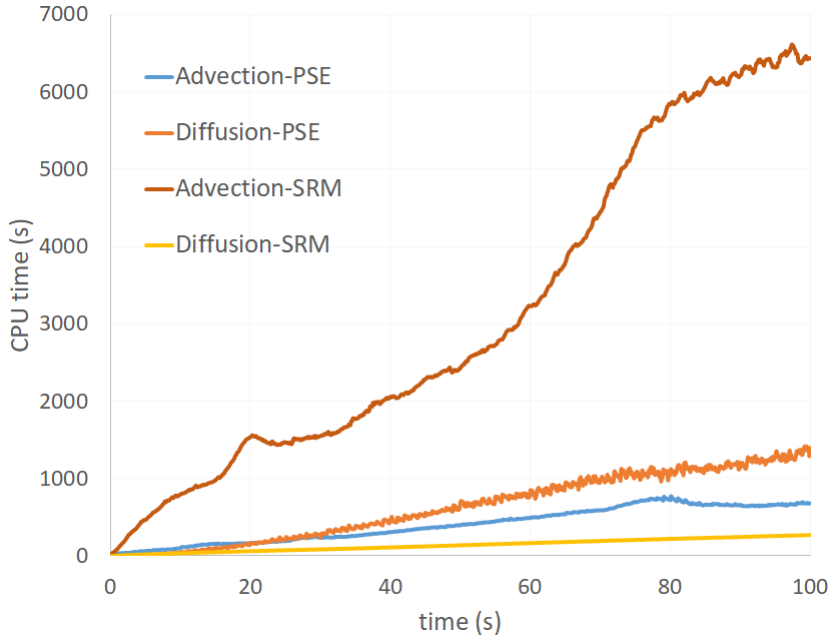

Figure 6.23: CPU time for the advection and diffusion step.

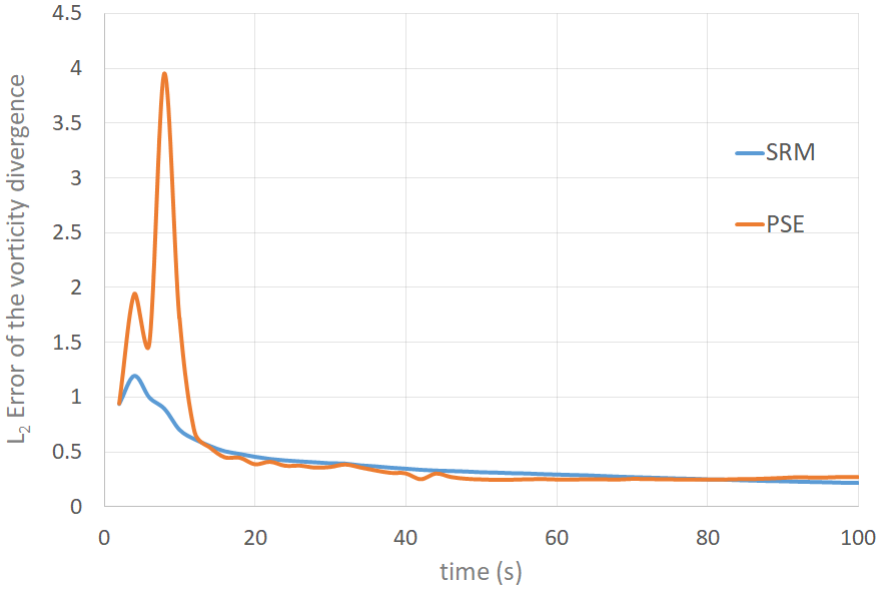

Figure 6.24: L_2 norm of the vorticity divergence vs time.

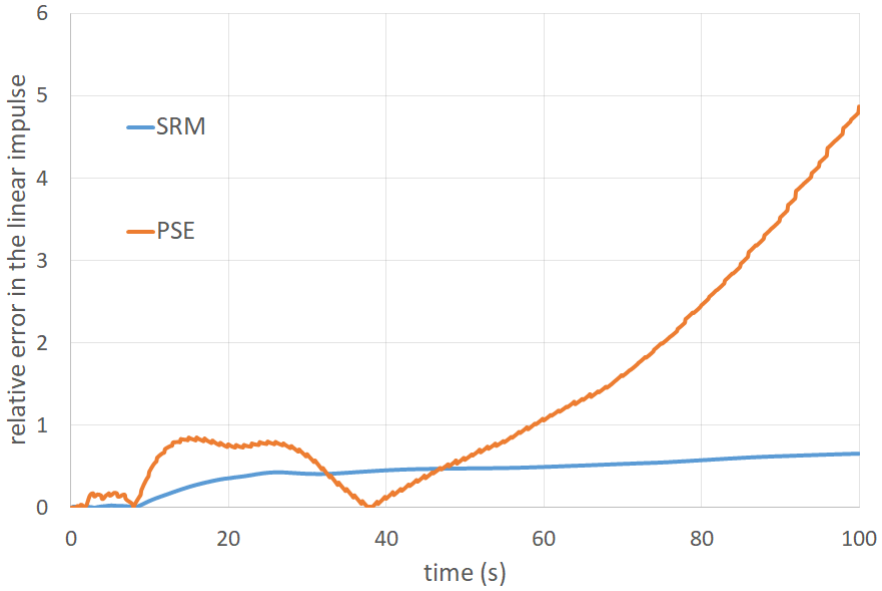

Figure 6.25: % relative error in the $z-$component of the total linear impulse vs time.

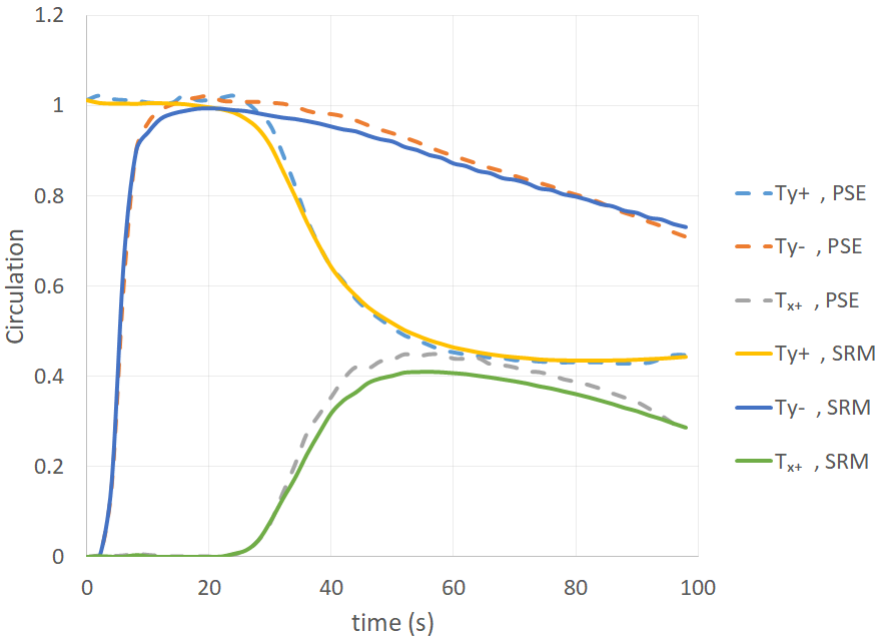

Figure 6.26: Time-development of circulation around interacting vortex tubes.

Chapter 7

Conclusion and Future Work

We presented simple expressions for approximating the velocity and vortex stretching vectors induced by a far-field collection of point vortices and we proposed a set of recurrence relations to properly evaluate these expressions over the entire range of polar angle. The expressions are used by an adaptive error-controlled hybrid fast solver that minimizes, for a given accuracy, the cost of approximating the velocity and vortex stretching vectors using a regularized representation of the vorticity field. The hybrid fast solver uses both multipole and local spherical harmonic based expansions to approximate the vector fields. The proposed solver extends Greengard [15] fast multipole algorithm to continuously optimize the speed of the fast multipole method to adapt to the evolving flow-field, while meeting a given accuracy constraint. The automated scheme employs, in addition to the order of expansion p^*, the parameters $n_T^*(p^*)$ and $n_F^*(p^*)$ to control the depth of the tree structure and to balance the far-field and near-field evaluations by properly choosing, based on well-defined criteria, the type of interaction (cell-cell, cell-element, and element-element) to perform in order to compute the contribution of non-neighboring cells. The scheme also uses a criterion, based on the critical tree depth d_σ^*, to limit the tree depth to keep the regularization error below the user defined error threshold. Optimal values of the parameters p^*, n_T^*, n_F^*, and d_σ^* are periodically evaluated by the automated algorithm to account for the evolving elements distribution. For a given implementation on a given architecture and a given choice of the regularization core function, the dependence of n_T^* and n_F^* on p is established in a one-time only preprocessing step.

To investigate the accuracy of the introduced expressions, we carried out several simulations where the convergence of the velocity and the vortex stretching vectors as a function of the expansion order p is inspected. These simulations showed that the solutions converge to the direct solution with the expansion order as C^p, where C ranges between 0.41 and 0.48. To assess the performance of the adaptive scheme, we carried out several simulations where the CPU time for evaluating both velocity and vortex stretching vectors is reported for different values of n_F and different values of the number of elements. These simulations showed that for values of n_F other than $n_{F,\sigma}^*(p^*)$, the adaptive scheme suffers from load imbalance between the near-field and far-field evaluations, resulting in a simulation time that oscillates above that of $n_{F,\sigma}^*$, where $n_{F,\sigma}^* = 129$ for $p^* = 5$. Element-element interactions dominate the cost for

$n_F > n^*_{F,\sigma}(p)$, while cell-cell interaction are dominant for the case of $n_F < n^*_{F,\sigma}(p)$, which is due to the associated cost of computing local expansions for cells in a deep tree structure. The simulations also showed that for an optimum value of n_F, the adaptive scheme scale as $\mathcal{O}(N^{1.045})$, hence assuming an almost linear dependence on the number of elements. Moreover, the simulations showed that the truncation error exponentially decreases as p increases, while the regularization error (and thus the total error) decreases in jumps. These jumps occur when the tree depth decreases in response to an increase in n^*. We concluded that for a regularized representation of the vorticity field, it is inefficient to increase the order of expansion p unless it is accompanied by a drop in the tree depth since this will only increase the computational cost without improving the accuracy. We showcased the adaptive error-controlled hybrid fast solver by simulating the collision of two vortex rings at Reynolds number $Re = 1000$ while constraining the maximum error on the average L_2 velocity error to a pre-defined total error of $E_T = 10^{-3}$. The results showed that the proposed scheme realizes optimal speed for a given accuracy by automatically computing the optimal values of the scheme parameters to adapt to the evolving flow-field.

The adaptive scheme, along with the derived expressions, are employed to simulate the collision of two vortex rings over a long period of time. In these simulations, an extension to three dimensions of the smoothed redistribution method is used to model the diffusion. The simulations were carried out for the following values of Reynolds number: 500, 1000, 1500, 2000. The performance of the method is then assessed by comparing the evolution of the rings with previous work. The simulations showed that the first reconnection, in which the two rings fuse into a single distorted ring, persists for all considered values of Re, which is in agreement with reported experiments. The rings undergo a second reconnection which is slower than the first reconnection and where the cancellation of vorticity is incomplete. The remnant vortices have higher circulation for larger Re and the second reconnection is nearly not existing for $Re = 1500$ and 2000. These findings are also in agreement with previous experiments. The simulations uncovered a third reconnection for the case $Re = 500$, which is not reported elsewhere. The convergence of the method is also inspected in term of conserving the following flow invariants: total circulation and linear and angular impulse. All the invariants were nearly conserved (98%). Despite the fact that the method does not implement an algorithm for nullifying the non-divergent component of the computed vorticity vector, the vorticity divergence free condition was maintained for the low Reynolds number cases. This is because the redistribution scheme injects elements in gaps, created by the large strain rates, in a manner that maintains the overlap condition between neighboring elements. For large values of Re, and for moderate values of the time step, the redistribution scheme was not able to fill the gaps properly and ensure overlap which caused a noticeable non-zero divergence in the vorticity field. In this case, reducing the time step decreases the ratio of the advection to injection lengths. This effectively increases the number of injected elements per unit time which acts to maintain the spatial resolution and dampen the variation of L_2 norm of $\nabla.\vec{\omega}$, as was observed during the first reconnection the $Re = 2000$ case.

We presented a comparative study of PSE and the 3D extension of the smoothed redistribution diffusion model (SRM) in simulating the collision of two vortex rings for

Reynolds number $Re = 500$. The simulations showed that similar results are obtained by the two diffusion models. In fact, both models were able to went through the "fusion" and "fission" process, and both schemes were able to identify a third vortex reconnection that took place at $t \simeq 98$. Moreover, both schemes manage to maintain a nearly divergence free vorticity field throughout the simulation. Although the linear invariants are better conserved in the SRM diffusion model, the overall CPU time (advection & diffision) for the SRM case is much larger than that of the PSE. This is related to the additional computational cost of advecting a very large number of elements in the SRM case since the number of elements introduced by the SRM model is about five time that introduced by the PSE model. The reason why we have obtained such a large number of elements by the SRM model is that we didn't apply any procedure to merge nearby elements or to redistribute elements with small strengths. The simulations also showed that the CPU time of the diffusion step of the PSE case is much larger than that of the PSE case, and it is even larger the CPU time of the advection step in the PSE case. In fact, the evaluation of the diffusion term in the PSE model requires $O(N^2)$ computational operation where N is the number of elements. Although we used a small cutoff diffusion distance of 3.2σ, the CPU time of the diffusion step is approximatively twice that of the advection step for the PSE case. Future work should implement and apply merging and redistribution procedure along with the SRM model and compare the two diffusion models for higher Reynolds number.
We also presented few relaxation schemes introduced to force the particle vorticity field to remain nearly divergence-free at all times. Upon investigating the conservation properties of the relaxation scheme introduced by Winckelmans, we showed that every time we apply this method to reconstruct a divergence free vorticity field, the total circulation and linear impulse are multiplied by a factor of $\frac{2}{3}$. Hence, this scheme produces non-physical results since it acts as an external net force that dissipates energy out of the system and slow down fluid particles. Future work concerns deeper analysis of some particular mechanisms, to maintain the divergence free condition without affecting the conserved quantities.

Bibliography

[1] A. J. Chorin, "Vortex methods," pam report 593, Department of Mathematics, University of Berkeley, 1993.

[2] A. Leonard, "Vortex methods for flow simulation," J. Comput. Phys., vol. 37, pp. 289–335, 1980.

[3] M. J. Stock, "Summary of vortex methods literature (a living document rife with opinion)," 2007.

[4] G.-H. Cottet and P. Koumoutsakos, Vortex Methods: Theory and Applications. Cambridge, UK: Cambridge Univ. Press, 2000.

[5] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart, "Vortex methods and their application to trailing wake vortex simulations," Comptes Rendus Physique, vol. 6, no. 4-5, pp. 467–486, 2005.

[6] R. Cocle, G. Winckelmans, and G. Daeninck, "Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations. submitted to j," Comp. Phys. in Ap, 2007.

[7] L. Rosenhead, "The formation of vortices from a surface of discontinuity," Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, vol. 134, no. 823, pp. 170–192, 1931.

[8] G. Birkhoff and J. Fisher, "Do vortex sheets roll up?," Rendiconti del Circolo matematico di Palermo, vol. 8, no. 1, pp. 77–90, 1959.

[9] A. J. Chorin and P. S. Bernard, "Discretization of a vortex sheet, with an example of roll-up," Journal of Computational Physics, vol. 13, no. 3, pp. 423–429, 1973.

[10] A. J. Chorin, "Numerical study of slightly viscous flow," J. Fluid Mech., vol. 57, pp. 785–796, 1973.

[11] S. Mas-Gallic and P. Raviart, "Particle approximation of convection-diffusion problems," Report of Univ de Paris, vol. 6, 1986.

[12] S. Shankar and L. Dommelen, "A new diffusion procedure for vortex methods," J. Comput. Phys., vol. 127, no. 1, pp. 88–109, 1996.

[13] I. Lakkis and A. Ghoniem, "Axisymmetric vortex method for low-mach number, diffusion-controlled combustion," J. Comput. Phys., vol. 182, no. 2, pp. 435–475, 2003.

[14] J. E. Barnes and P. Hut, "A hierarchical $o(nlogn)$ force calculations algorithm," Nature, vol. 324, p. 446, 1986.

[15] L. Greengard and V. Rohklin, "A fast algorithm for particle simulations,," J. Comput. Phys., vol. 73, p. 325, 1987.

[16] A. J. Chorin, "Vortex sheed approximations of boundary layers," J. Comput. Phys., vol. 27, pp. 428–442, 1978.

[17] P. Koumoutsakos, A. Leonard, and F. Pepin, "Boundary conditions for viscous vortex methods," J. Comput. Phys., vol. 113, pp. 52–61, 1994.

[18] J. Marshall and J. Grant, "Penetration of a blade into a vortex core: vorticity response and unsteady blade forces," Journal of Fluid Mechanics Digital Archive, vol. 306, pp. 83–109, 1996.

[19] I. Lakkis, "Grid-free vortex methods for natural convection; handling source terms and nonlinear diffusion," Numerical Heat Transfer Part B: Fundamentals, vol. 62, no. 5, pp. 370–398, 2012.

[20] R. Krasny, "Desingularization of periodic vortex sheet roll-up," Journal of Computational Physics, vol. 65, no. 2, pp. 292–313, 1986.

[21] R. Krasny, "A study of singularity formation in a vortex sheet by the point-vortex approximation," Journal of Fluid Mechanics, vol. 167, pp. 65–93, 1986.

[22] A. Leonard, D. Shiels, J. K. Salmon, G. S. Winckelmans, and P. Ploumhans, "Recent advances in high resolution vortex methods for incompressible flows," 13th AIAA Computational Fluid Dynamics Conf., vol. 97, p. 2108, June 29–July 2 1997.

[23] P. Ploumhans and G. S. Winckelmans, "Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry," Journal of Computational Physics, vol. 165, pp. 354–406, 2000.

[24] B. Hakizumwami, "High reynolds number flow past an impulsively started circular cylinder," Computers & Fluids, vol. 23, no. 7, pp. 895–902, 1994.

[25] A. Cheer, "Unsteady separated wake behind an impulsively started cylinder in slightly viscous flow.," J. Fluid. Mech., vol. 201, pp. 485–505, 1989.

[26] O. Knio and G. A. F., "3-dimensional vortex simulation of rollup and entrainment in a shear-layer," J. Comput. Phys., vol. 97, pp. 172–223, 1991.

[27] C. Anderson and C. Greengard, "On vortex methods," SIAM journal on numerical analysis, vol. 22, no. 3, pp. 413–440, 1985.

[28] L. Greengard, V. Rokhlin, C. Anderson, and C. Greengard, "Vortex methods," 1988.

[29] O. Knio and A. Ghoniem, "Vortex simulation of a three-dimensional reacting shear layer with infinite-rate kinetics," AIAA journal, vol. 30, no. 1, pp. 105–116, 1992.

[30] M. Soteriou and A. Ghoniem, "Numerical simulation of unsteady combustion using the transport element method," in ESAIM: Proceedings, vol. 1, pp. 429–446, Citeseer, 1996.

[31] A. Krishnan and A. Ghoniem, "Simulation of rollup and mixing in rayleigh-taylor flow using the transport-element method," Journal of Computational Physics, vol. 99, pp. 1–27, 1992.

[32] I. Lakkis and A. F. Ghoniem, "Lagrangian simulation of fire plumes," 7th AIAA/ASME Joint Thermo. and Heat Transf. Conf., vol. 1, pp. 215–226, 1998.

[33] A. F. Ghoniem, I. Lakkis, and S. M., "Numerical simulation of the dynamics of large fire plumes and the phenomenon of puffing," 26th Symp. (Int.) Comb. The Combustion Institute, pp. 1531–1539, 1996.

[34] J. D. Eldredge, T. Colonius, and A. Leonard, "A vortex particle method for two-dimensional compressible flow," Journal of Computational Physics, vol. 179, no. 2, pp. 371 – 399, 2002.

[35] F. Schlegel, D. Wee, and A. Ghoniem, "A fast 3d particle method for the simulation of buoyant flow," Journal of Computational Physics, vol. 227, no. 21, pp. 9063–9090, 2008.

[36] J. Ray, H. Najm, R. Milne, K. Devine, and S. Kempka, "Triple flame structure and dynamics at the stabilization point of an unsteady lifted jet diffusion flame," PROCEEDINGS-COMBUSTION INSTITUTE, vol. 28, no. 1, pp. 219–226, 2000.

[37] H. Najm, R. Milne, K. Devine, and S. Kempka, "A coupled lagrangian-eulerian scheme for reacting flow modeling," in ESAIM: Proceedings, vol. 7, pp. 304–313, edpsciences. org, 1999.

[38] I. Lakkis and A. Ghoniem, "Grid-free simulation of radiative transport in a participating medium," in ESAIM: Proceedings, vol. 7, pp. 234–246, edpsciences. org, 1999.

[39] I. Lakkis and A. Ghoniem, "Grid-free simulation of a reacting radiating fuel ring," in AIAA, Aerospace Sciences Meeting and Exhibit, 38 th, Reno, NV, 2000.

[40] I. Lakkis and A. Ghoniem, "A high resolution spatially adaptive vortex method for separating flows. part i: Two-dimensional domains," J. Comput. Phys., vol. 228, no. 2, pp. 491–515, 2009.

[41] M. Ould-Salihi, G. Cottet, and M. El Hamraoui, "Blending finite-difference and vortex methods for incompressible flow computations," SIAM Journal on Scientific Computing, vol. 22, no. 5, pp. 1655–1674, 2001.

[42] E. Stein, R. De Borst, and T. J. Hughes, "Encyclopedia of computational mechanics," 2004.

[43] G. Winckelmans, U. C. D. Louvain, J. K. Salmon, A. Leonard, and B. Jodoin, "Application of fast parallel and sequential tree codes to computing three-dimensional flows with the vortex element and boundary element methods," 1996.

[44] G. Pedrizzetti, "Insight into singular vortex flows," Fluid dynamics research, vol. 10, no. 2, p. 101, 1992.

[45] J. Christiansen, "Numerical simulation of hydrodynamics by the method of point vortices," Journal of Computational Physics, vol. 13, no. 3, pp. 363–379, 1973.

[46] G. Cottet, B. Michaux, S. Ossia, and G. VanderLinden, "A comparison of spectral and vortex methods in three-dimensional incompressible flows," Journal of Computational Physics, vol. 175, no. 2, pp. 702–712, 2002.

[47] P. Chatelain and P. Koumoutsakos, "A fourier-based elliptic solver for vortical flows with periodic and unbounded directions," Journal of Computational Physics, vol. 229, no. 7, pp. 2425–2431, 2010.

[48] R. Yokota, "An FMM based on dual tree traversal for many-core architectures," CoRR, vol. abs/1209.3516, 2012.

[49] R. Yokota and L. Barba, "Hierarchical n-body simulations with autotuning for heterogeneous systems," Computing in Science Engineering, vol. 14, pp. 30–39, May 2012.

[50] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren, "Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at re = 300, 500, and 1000," J. Comput. Phys., vol. 178, pp. 427–463, 2002.

[51] R. Yokota, T. K. Sheel, and S. Obi, "Calculation of isotropic turbulence using a pure lagrangian vortex method," Journal of Computational Physics, vol. 226, no. 2, pp. 1589–1606, 2007.

[52] R. Yokota, T. Narumi, R. Sakamaki, S. Kameoka, S. Obi, and K. Yasuoka, "Fast multipole methods on a cluster of gpus for the meshless simulation of turbulence," Computer Physics Communications, vol. 180, no. 11, pp. 2066–2078, 2009.

[53] R. Yokota and L. Barba, "Fmm-based vortex method for simulation of isotropic turbulence on gpus, compared with a spectral method," Computers & Fluids, vol. 80, pp. 17–27, 2013.

[54] T. Berdowski, J. Walther, C. Ferreira, and F. Meng, "Derivation and analysis of the analytical velocity and vortex stretching expressions for an o (n log n)-fmm," in Journal of Physics: Conference Series, vol. 753, p. 082023, IOP Publishing, 2016.

[55] J. Beale and A. Majda, "High order accurate vortex methods with explicit velocity kernels," Journal of Computational Physics, vol. 58, no. 2, pp. 188–208, 1985.

[56] G. Winckelmans and A. Leonard, "Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows," Journal of Computational Physics, vol. 109, no. 2, pp. 247–273, 1993.

[57] R. Beatson and L. Greengard, "A short course on fast multipole methods," Wavelets, multilevel methods and elliptic PDEs, vol. 1, pp. 1–37, 1997.

[58] W. Bosch, "On the computation of derivatives of legendre functions," Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy, vol. 25, no. 9-11, pp. 655–659, 2000.

[59] M. Petrovskaya and A. Vershkov, "Non-singular expressions for the gravity gradients in the local north-oriented and orbital reference frames," Journal of Geodesy, vol. 80, no. 3, pp. 117–127, 2006.

[60] W. Dehnen, "A very fast and momentum-conserving tree code," The Astrophysical journal, vol. 536, pp. L39–L42, 07 2000.

[61] W. Dehnen, "A hierarchical o (n) force calculation algorithm," arXiv preprint astro-ph/0202512, 2002.

[62] R. Yokota and L. Barba, "Parameter tuning of a hybrid treecode-fmm on gpus," 2011.

[63] M. S. Warren and J. K. Salmon, "A portable parallel particle program," Computer Physics Communications, vol. 87, no. 1, pp. 266 – 290, 1995. Particle Simulation Methods.

[64] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," Journal of Computational Physics, vol. 155, no. 2, pp. 468 – 498, 1999.

[65] R. Capuzzo-Dolcetta and P. Miocchi, "A comparison between the fast multipole algorithm and the tree-code to evaluate gravitational forces in 3-d," Journal of Computational Physics, vol. 143, no. 1, pp. 29–48, 1998.

[66] R. Yokota and L. A. Barba, "Fast n-body simulations on gpus," CoRR, abs/1108.5815, 2011.

[67] S. Salloum and I. Lakkis, "Proper evaluation of spherical harmonics-based expressions for the velocity and vortex stretching vectors in three-dimensional grid-free vortex methods," Journal of Computational Physics, p. 109603, 2020.

[68] S. Kida, M. Takaoka, and F. Hussain, "Collision of two vortex rings," Journal of Fluid Mechanics, vol. 230, pp. 583–646, 1991.

[69] T. Kambe and T. Takao, "Motion of distorted vortex rings," Journal of the Physical Society of Japan, vol. 31, no. 2, pp. 591–599, 1971.

[70] T. Fohl and J. Turner, "Colliding vortex rings," The Physics of Fluids, vol. 18, no. 4, pp. 433–436, 1975.

7.1 Appendix A: Error bounds for the potential and velocity vectors

7.1.1 Error Bounds for the Potential Vector Induced by a Singular Source Element

Consider a singular source element j of strength $\vec{\alpha}_j$ located at $Q = (\rho_j, \theta_j, \varphi_j)$. The potential vector induced by j at a target element located at $P = (r, \theta, \varphi)$ has the following expression:

$$\vec{\psi}_j(P) = \frac{\vec{\alpha}_j}{4\pi d_j}, \tag{7.1}$$

where $d_j = \parallel \vec{QP} \parallel$ is the distance between source and target elements, as depicted in Figure 7.1. Let γ be the angle between the vectors P and Q. From the law of cosines,

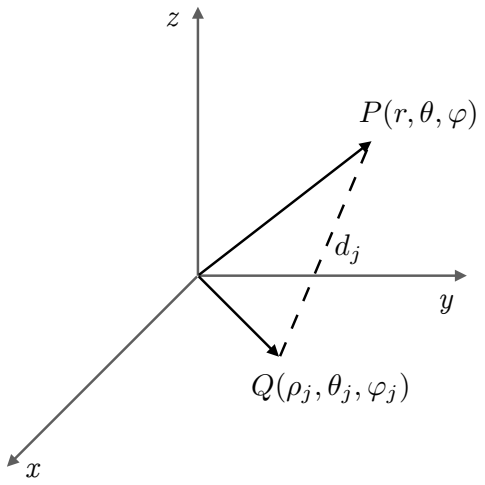

Figure 7.1: Schematic showing source and target.

we have:

$$d_j^2 = r^2 + \rho_j^2 - 2r\rho_j cos\gamma \tag{7.2}$$

From this relation, we may write

$$\vec{\psi}_j(P) = \frac{\vec{\alpha}_j}{4\pi r} \frac{1}{\sqrt{1 - 2\left(\frac{\rho_j}{r}\right) cos\gamma + \left(\frac{\rho_j}{r}\right)^2}} \tag{7.3}$$

Let $\mu_j = \frac{\rho_j}{r}$. For $\rho_j < r$, that is $\mu_j < 1$, if we expand $\vec{\psi}(P)$ in term of Legendre Polynomial we obtain:

$$\vec{\psi}_j(P) = \frac{\vec{\alpha}_j}{4\pi r} \sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \tag{7.4}$$

The error resulting from truncating $\vec{\psi}_j(P)$ at some order p is:

$$E_{\vec{\psi}_j} = \left|\left| \frac{\vec{\alpha}_j}{4\pi d_j} - \frac{\vec{\alpha}_j}{4\pi r}\sum_{n=0}^{p} \mu_j^n P_n(cos\gamma) \right|\right| = \left|\left| \frac{\vec{\alpha}_j}{4\pi r}\sum_{n=p+1}^{\infty} \mu_j^n P_n(cos\gamma) \right|\right| \tag{7.5}$$

Knowing that $P_n(cos\gamma) \leq 1$, and using the triangle inequality, we obtain an error bound for the multipole expansion

$$E_{\vec{\psi}_j} \leq \left|\left| \frac{\vec{\alpha}_j}{4\pi r}\sum_{n=p+1}^{\infty} \mu_j^n \right|\right| \leq \frac{\parallel \vec{\alpha_j} \parallel}{4\pi r}\frac{\mu_j^{p+1}}{1-\mu_j} \tag{7.6}$$

$$E_{\vec{\psi}_j} \leq \frac{1}{4\pi}\frac{\parallel \vec{\alpha_j} \parallel}{(r-\rho_j)}\left(\frac{\rho_j}{r}\right)^{p+1} \tag{7.7}$$

Similary, when $r < \rho_j$, we define $\mu_j = \frac{r}{\rho_j}$. The potential vector and the error bound for the local expansion would have the following expressions:

$$\vec{\psi}_j(P) = \frac{\vec{\alpha}_j}{4\pi d_j} = \frac{\vec{\alpha}_j}{4\pi\rho_j}\sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \tag{7.8}$$

$$E_{\vec{\psi}_j} \leq \frac{1}{4\pi}\frac{\parallel \vec{\alpha_j} \parallel}{(\rho_j-r)}\left(\frac{r}{\rho_j}\right)^{p+1} \tag{7.9}$$

7.1.2 Error Bounds for the Potential Vector Induced by a Cluster of Source Elements

Suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located at the points $\vec{Q}_j = (\rho_j, \theta_j, \varphi_j)$ inside the sphere D_Q of radius a with center at $Q = (0,0,0)$, then at any $\vec{P} = (r, \theta, \varphi)$ with $r > a$, the potential vector field is given by:

$$\vec{\psi} = \sum_j \vec{\psi}_j = \sum_j \frac{\vec{\alpha}_j}{4\pi d_j} = \sum_j \frac{\vec{\alpha}_j}{4\pi r}\sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \tag{7.10}$$

The error resulting from truncating $\vec{\psi}(P)$ at some order p is:

$$E_{\vec{\psi}} = \left|\left| \sum_j \left(\frac{\vec{\alpha}_j}{4\pi d_j} - \frac{\vec{\alpha}_j}{4\pi r}\sum_{n=0}^{p} \mu_j^n P_n(cos\gamma) \right) \right|\right| \tag{7.11}$$

Using the triangle inequality, and since $\mu_j \leq \frac{a}{r}$, the multipole expansion error will be bounded by:

$$E_{\vec{\psi}} \leq \frac{1}{4\pi}\frac{\Gamma_{D_Q}}{(r-a)}\left(\frac{a}{r}\right)^{p+1} \tag{7.12}$$

where $\Gamma_{D_Q} = \sum_j \parallel \vec{\alpha_j} \parallel$

Now suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located inside a cubic box J_l at level l with center at $Q = (0, 0, 0)$. The radius of the smallest sphere which encloses box J_l is $a_{J_l} = \frac{\sqrt{3}}{2} W_{J_l}$ where W_{J_l} is the box width. Following the adaptive solver Scheme described in section 3.1, the distance between the center of box J_l and any point $P = (r, \theta, \varphi)$ located outside the neighborhood of box J_l is $r_{J_l} \geq (n_D + 0.5) W_{J_l}$. For any $p \geq 1$, the multipole expansion error will be bounded by:

$$\vec{E}_{M,\vec{\psi}} \leq \frac{1}{4\pi} \frac{\Gamma_{J_l}}{W_{J_l}} \frac{1}{n_D - \frac{\sqrt{3}-1}{2}} \left(\frac{\sqrt{3}}{2n_D + 1} \right)^{p+1} \tag{7.13}$$

Noting that $W_{J_l} = W_0 / 2^l$, we get

$$E_{M,\vec{\psi}} \leq \frac{1}{4\pi} \frac{\Gamma_{J_l}}{W_0} \frac{2^l}{n_D - \frac{\sqrt{3}-1}{2}} \left(\frac{\sqrt{3}}{2n_D + 1} \right)^{p+1} \tag{7.14}$$

where W_0 is the width of the root box, $\Gamma_{J_l} = \sum_j \parallel \vec{\alpha_j} \parallel$ and the summation is done over all the source elements within box J_l.

Similary, suppose that s vortices with strength $(\vec{\alpha}_i, i = 1 \ldots s)$ are located inside the sphere D_Q of radius a with center at $Q = (\rho, \alpha, \beta)$, and that $\rho = (c + 1)a$ with $c > 1$, then for any target $P = (r, \theta, \phi)$ inside the sphere D_0 of radius a centred at the Origin, the potential vector is given by

$$\vec{\psi} = \sum_j \vec{\psi}_j = \sum_j \frac{\vec{\alpha}_j}{4\pi d_j} = \sum_j \frac{\vec{\alpha}_j}{4\pi \rho_j} \sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \tag{7.15}$$

The error resulting from truncating $\vec{\psi}(P)$ at some order p is:

$$E_{\vec{\psi}} = \left\| \sum_j \left(\frac{\vec{\alpha}_j}{4\pi d_j} - \frac{\vec{\alpha}_j}{4\pi \rho_j} \sum_{n=0}^{p} \mu_j^n P_n(cos\gamma) \right) \right\| \tag{7.16}$$

Since $\mu_j = \frac{r}{\rho_j} \leq \frac{a}{\rho - a} = \frac{1}{c}$, using the triangle inequality, the local expansion error will be bounded by:

$$E_{\vec{\psi}} \leq \frac{1}{4\pi} \frac{\Gamma_{D_Q}}{(ca - a)} \left(\frac{1}{c} \right)^{p+1} \tag{7.17}$$

where $\Gamma_{D_Q} = \sum_j \parallel \vec{\alpha_j} \parallel$.

Now suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located inside a cubic box J_l at level l with center at $Q = (\rho, \alpha, \beta)$. The radius of the smallest sphere which encloses box J_l is $a_{J_l} = \frac{\sqrt{3}}{2} W_{J_l}$ where W_{J_l} is the box width. Following the adaptive solver Scheme described in section 3.1, an upper bound for c is given by

$$c \geq \frac{2\sqrt{3}}{3}(n_D + 1) - 1, \tag{7.18}$$

and for any $p \geq 1$, the local expansion error that results upon approximating the potential vector at any point P within a box I_l centered at the origin is bounded by:

$$E_{L,\vec{\psi}} \leq \frac{1}{4\pi} \frac{\Gamma_{J_l}}{W_0} \frac{2_l}{n_D + 1 - \frac{\sqrt{3}}{2}} \left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D+1) - 1} \right)^{p+1} \tag{7.19}$$

7.1.3 Error Bounds for the Velocity Vector Induced by a Singular Source Element

Once again, let us consider a singular source element j of strength $\vec{\alpha}_j$ located at $Q = (\rho_j, \theta_j, \varphi_j)$. The velocity at a target element $P = (r, \theta, \varphi)$ is calculated as the curl of the potential vector $\vec{\psi}_j$

$$\vec{u}_j(P) = \nabla \times \vec{\psi}_j(P) = \nabla \times \left(\frac{\vec{\alpha}_j}{4\pi r} \sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \right) = \sum_{n=0}^{\infty} \frac{\rho_j^n}{4\pi} \nabla \left(\frac{P_n(cos\gamma)}{r^{n+1}} \right) \times \vec{\alpha}_j \tag{7.20}$$

The error resulting from truncating $\vec{u}_j(P)$ at some order p is:

$$E_{\vec{u}_j} = \left\| \sum_{n=p+1}^{\infty} \frac{\rho_j^n}{4\pi} \nabla \left(\frac{P_n(cos\gamma)}{r^{n+1}} \right) \times \vec{\alpha}_j \right\| \tag{7.21}$$

Since $\| \vec{u} \times \vec{v} \| \leq \| \vec{u} \| \| \vec{v} \|$ for any two vectors $\vec{u}$ and $\vec{v}$, using the triangle inequality, the truncating error will be bounded by:

$$E_{\vec{u}_j} \leq \frac{\| \vec{\alpha}_j \|}{4\pi} \sum_{n=p+1}^{\infty} \rho_j^n \left\| \nabla \left(\frac{P_n(cos\gamma)}{r^{n+1}} \right) \right\| \tag{7.22}$$

Since the norm of a vector does not vary when we change the reference frame, we will evaluate the expression of $\left\| \nabla \left(\frac{P_n(cos\gamma)}{r^{n+1}} \right) \right\|$ in a new frame obtained by rotating the old frame such that the source element Q will be locatated on the new zz' axe, that is we have $cos\gamma = cos\theta$

$$\nabla \left(\frac{P_n(cos\theta)}{r^{n+1}} \right) = -\frac{1}{r^{n+2}} \left((n+1) P_n(cos\theta), sin(\theta) {P'}_n(cos\theta), 0 \right)^T \tag{7.23}$$

$$\left\| \nabla \left(\frac{P_n(cos\theta)}{r^{n+1}} \right) \right\| = \frac{1}{r^{n+2}} \sqrt{(n+1)^2 P_n^2(cos\theta) + sin^2(\theta) {P'}_n^2(cos\theta)} \tag{7.24}$$

Knowing that

$$(n+1)^2 P_n^2(x) + (1-x^2) {P'}_n^2(x) \leq (n+1)^2 \tag{7.25}$$

we obtain :

$$\left\| \nabla \left(\frac{P_n(cos\theta)}{r^{n+1}} \right) \right\| \leq \frac{n+1}{r^n + 2} \tag{7.26}$$

Replacing in equation (7.22), the multipole expansion error of the velocity vector is bounded by:

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi r^2} \sum_{n=p+1}^{\infty} (n+1)\mu_j^n \tag{7.27}$$

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi r^2} \left[\frac{\mu_j^{p+1}}{1-\mu_j} + \mu \frac{d}{d\mu_j} \left(\frac{\mu_j^{p+1}}{1-\mu_j} \right) \right] \tag{7.28}$$

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi r^2} \frac{\mu_j^{p+1}}{(1-\mu_j)^2} \left[p+2-\mu_j\,(p+1) \right] \tag{7.29}$$

$$E_{\vec{u}_j} \leq \frac{1}{4\pi} \frac{\parallel \vec{\alpha}_i \parallel}{(r-\rho_j)^2} \left(\frac{\rho_j}{r} \right)^{p+1} \left[p+2-\frac{\rho_j}{r}\,(p+1) \right] \tag{7.30}$$

Similary, when $r < \rho_j$, $\mu_j = \frac{r}{\rho_j}$, the velocity vector induced by j at a target $P = (r, \theta, \phi)$ is calculated as the curl of the potential vector and has the following expression:

$$\vec{u}_j(P) = \nabla \times \vec{\psi}_j(P) = \nabla \times \left(\frac{\vec{\alpha}_j}{4\pi\rho_j} \sum_{n=0}^{\infty} \mu_j^n P_n(cos\gamma) \right) = \sum_{n=0}^{\infty} \frac{1}{4\pi\rho_j^n} \nabla \left(r^n P_n(cos\gamma) \right) \times \vec{\alpha}_j \tag{7.31}$$

The error resulting from truncating $\vec{u}_j(P)$ at some order p is:

$$E_{\vec{u}_j} = \left\| \sum_{n=p+1}^{\infty} \frac{1}{4\pi\rho_j^n} \nabla \left(r^n P_n(cos\gamma) \right) \times \vec{\alpha}_j \right\| \tag{7.32}$$

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi} \sum_{n=p+1}^{\infty} \frac{1}{\rho_j^n} \left|\left| \nabla \left(r^n P_n(cos\gamma) \right) \right|\right| \tag{7.33}$$

Once again, the norm of a vector does not vary when we change the reference frame. Calculating the norm in the new reference frame decribed above, we obtain:

$$\nabla \left(r^n P_n(cos\gamma) \right) = r^{n-1} (n P_n(cos\theta), -sin\theta P'_n(cos\theta), 0)^T \tag{7.34}$$

$$||r^n P_n(cos\gamma)|| = r^{n-1} \sqrt{n^2 P_n^2(cos\theta) + sin^2\theta P'^2_n(cos\theta)} \tag{7.35}$$

Knowing that

$$n^2 P_n^2(x) + (1-x^2) P'^2_n(x) \leq n^2 \tag{7.36}$$

we obtain :

$$||r^n P_n(cos\gamma)|| = n r^{n-1} \tag{7.37}$$

Replacing in equation (7.33), the local expansion error of the velocity vector is bounded by:

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi\rho_j} \sum_{n=p+1}^{\infty} n\mu_j^n \tag{7.38}$$

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi\rho_j}\mu\frac{d}{d\mu_j}\left(\frac{\mu_j^{p+1}}{1-\mu_j}\right) \tag{7.39}$$

$$E_{\vec{u}_j} \leq \frac{\parallel \vec{\alpha}_j \parallel}{4\pi\rho_j^2}\frac{\mu_j^{p+1}}{(1-\mu_j)^2}\left[p+2-\mu_j p\right] \tag{7.40}$$

$$E_{\vec{u}_j} \leq \frac{1}{4\pi}\frac{\parallel \vec{\alpha}_i \parallel}{(\rho_j - r)^2}\left(\frac{r}{\rho_j}\right)^{p+1}\left[p+2-\frac{r}{\rho_j}p\right] \tag{7.41}$$

7.1.4 Error Bounds for the Velocity Vector Induced by a Cluster of Source Elements

Suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located at the points $\vec{Q}_j = (\rho_j, \theta_j, \varphi_j)$ inside the sphere D_Q of radius a with center at $Q = (0,0,0)$, then at any $\vec{P} = (r, \theta, \varphi)$ with $r > a$, the velocity vector field is given by:

$$\vec{u} = \sum_j \vec{u}_j = \sum_j \sum_{n=0}^{\infty} \frac{\rho_j^n}{4\pi}\nabla\left(\frac{P_n(cos\gamma)}{r^{n+1}}\right) \times \vec{\alpha}_j \tag{7.42}$$

Using the triangle inequality, and since $\mu_j \leq \frac{a}{r}$, the multipole expansion error will be bounded by:

$$E_{\vec{u}} \leq \sum_i E_{\vec{u}_j} \leq \frac{1}{4\pi}\frac{\Gamma_{D_Q}}{(r-a)^2}\left(\frac{a}{r}\right)^{p+1}\left[p+2-\frac{a}{r}(p+1)\right] \tag{7.43}$$

where $\Gamma_{D_Q} = \sum_j \parallel \vec{\alpha_j} \parallel$.

Now suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located inside a cubic box J_l at level l with center at $Q = (0,0,0)$. For any $p \geq 1$, the multipole expansion error that results upon approximating the velocity vector at any point P outside the neighbours of J_l is bounded by:

$$E_{M,\vec{u}} \leq \frac{1}{4\pi}\frac{\Gamma_{J_l}}{W_0^2}\frac{4^l}{\left(n_D - \frac{\sqrt{3}-1}{2}\right)^2}\left(\frac{\sqrt{3}}{2n_D+1}\right)^{p+1}\left[p+2-\frac{\sqrt{3}}{2n_D+1}(p+1)\right] \tag{7.44}$$

Similary, suppose that s vortices having circulation $(\vec{\alpha}_i, i = 1 \ldots s)$ are located inside the sphere D_Q of radius a with center at $Q = (\rho, \alpha, \beta)$, and that $\rho = (c+1)a$ with $c > 1$, then for any target $P = (r, \theta, \phi)$ inside the sphere D_0 of radius a centred at the Origin, the velocity vector is given by

$$\vec{u} = \sum_j \vec{u}_j = \sum_j \sum_{n=0}^{\infty} \frac{1}{4\pi\rho_j^n}\nabla\left(r^n P_n(cos\gamma)\right) \times \vec{\alpha}_j \tag{7.45}$$

Since $\mu_j = \frac{r}{\rho_j} \leq \frac{a}{\rho - a} = \frac{1}{c}$, using the triangle inequality, the local expansion error will be bounded by:

$$E_{\vec{u}} \leq \sum E_{\vec{u}_j} \leq \frac{1}{4\pi} \frac{\Gamma_{D_Q}}{(ca-c)^2} \left(\frac{1}{c}\right)^{p+1} \left[p + 2 - \frac{1}{c}p\right] \tag{7.46}$$

where $\Gamma_{D_Q} = \sum_j \parallel \vec{\alpha_j} \parallel$.

Now suppose that s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ are located inside a cubic box J_l at level l with center at $Q = (\rho, \alpha, \beta)$. For any $p \geq 1$, the local expansion error that results upon approximating the velocity vector at any point P within a box I_l centered at the origin is bounded by:

$$E_{L,\vec{u}} \leq \frac{1}{4\pi} \frac{\Gamma_{J_l}}{W_0^2} \frac{4^l}{\left(n_D + 1 - \sqrt{3}\right)^2} \left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D+1) - 1}\right)^{p+1} \left[p + 2 - \left(\frac{1}{\frac{2\sqrt{3}}{3}(n_D+1) - 1}\right) p\right] \tag{7.47}$$

7.1.5 Regularization Error Bounds for the Potential Vector

In the case of regularized vortex method, an additional error $E_{\sigma,\vec{\psi}}$ aroses since G_σ deviates from $\frac{1}{4\pi r}$. Let us consider s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ located inside a cubic box J_l. Introducing the kernel $\beta_\sigma(r) = \frac{1}{\sigma}\beta\left(\frac{r}{\sigma}\right)$, and $\beta(r) = |G(r) - \frac{1}{4\pi r}|$ we obtain:

$$\begin{aligned} E_{\sigma,\vec{\psi}} &= \| \sum_{j \in J_l} \vec{\alpha}_j \left(G_\sigma(r_j) - \frac{1}{4\pi r_j} \right) \| \\ &\leq \sum_{j \in J_l} \|\vec{\alpha}_j\| |G_\sigma(r_j) - \frac{1}{4\pi r_j}| \\ &\leq |G_\sigma(r_m) - \frac{1}{4\pi r_m}| \sum_{j \in J_l} \|\vec{\alpha}_j\| \\ &\leq \Gamma_{J_l} \beta_\sigma(r_m) \end{aligned} \tag{7.48}$$

where $\Gamma_{J_l} = \sum_j \|\vec{\alpha}_j\|$, r_j is the distance between source element j and P, r_m is the distance that maximizes the kernel $\beta_\sigma(r)$. r^* depends only on the choice of the core function and for most cases we have $r_m = (n_D + 0.5)W_{min} = (n_D + 0.5)\frac{W_0}{2^l}$.

7.1.6 Regularization Error Bounds for the Velocity Vector

The regularizing error for the velocity arose since $q_\sigma(r)$ deviates from $\frac{1}{4\pi}$. Let us consider s vortices with strength $(\vec{\alpha}_j, j = 1 \ldots s)$ located inside a cubic box J_l. For any target element i we have:

$$
\begin{aligned}
E_{\sigma,\vec{u}} &= \|\sum_{j\in J_l} \frac{1}{4\pi r_j^3}(\vec{x_i}-\vec{x}_j)\wedge\vec{\alpha_j} - \frac{q_\sigma(r_j)}{r_j^3}(\vec{x_i}-\vec{x}_j)\wedge\vec{\alpha_j}\| \\
&\leq \sum_{j\in J_l} \|\frac{1}{4\pi r_j^3}(\vec{x_i}-\vec{x}_j)\wedge\vec{\alpha_j} - \frac{q_\sigma(r_j)}{r_j^3}(\vec{x_i}-\vec{x}_j)\wedge\vec{\alpha_j}\| \\
&\leq \sum_{j\in J_l} \frac{1}{4\pi r_j^3}|1-4\pi q_\sigma(r_j)| \parallel (\vec{x_i}-\vec{x}_j)\wedge\vec{\alpha_j} \parallel \\
&\leq \sum_{j\in J_l} \frac{1}{4\pi r_j^2}|1-4\pi q_\sigma(r_j)| \parallel \vec{\alpha_j} \parallel \\
&\leq \frac{\Gamma_{J_l}}{4\pi {r_m}^2}|1-4\pi q(\frac{r_m}{\sigma})| \\
&\leq \Gamma_{J_l}\kappa_\sigma(r_m)
\end{aligned} \tag{7.49}
$$

where $\Gamma_{J_l} = \sum_j \|\vec{\alpha}_j\|$, $\kappa(r) = \frac{1}{4\pi r^2}|1-4\pi q(r)|$ and $\kappa_\sigma(r) = \frac{1}{\sigma^2}\kappa(\frac{r}{\sigma})$, r_j is the distance between source element and target elements, r_m is the distance that maximizes the kernel $\kappa_\sigma(r)$.

7.2 Appendix B: Expressions of the Total Circulation, Linear and Angular Impulse, for a Regularized Vortex Method with Radially Symmetric Cutoff Function

$$\vec{\omega}_\sigma = \sum_p \vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) \tag{7.50}$$

$$\vec{\Gamma} = \int \vec{\omega}_\sigma dv = \int \sum_p \vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv = \sum_p \vec{\alpha}_p \int \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv = \sum_p \vec{\alpha}_p \tag{7.51}$$

$$\vec{I} = \int \vec{x}\times\vec{\omega}_\sigma dv = \int \vec{x}\times\sum_p \vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv = \sum_p \int \vec{x}\times\vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv \tag{7.52}$$

$$\vec{I} = \sum_p \int \vec{x}_p\times\vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv + \sum_p \int \left(\vec{x}-\vec{x}_p\right)\times\vec{\alpha}_p \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv \tag{7.53}$$

$$\vec{I} = \sum_p \vec{x}_p\times\vec{\alpha}_p \int \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv - \sum_p \vec{\alpha}_p \times \int \left(\vec{x}-\vec{x}_p\right) \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dv \tag{7.54}$$

$$\vec{I} = \sum_p \vec{x}_p\times\vec{\alpha}_p \tag{7.55}$$

$$\vec{A} = \int \vec{x}\times(\vec{x}\times\vec{\omega}_\sigma)dv = \int \vec{x}\times\left(\vec{x}\times\sum_p \vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv = \sum_p \int \vec{x}\times(\vec{x}\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right))\,dv \tag{7.56}$$

$$\vec{A} = \sum_p \int \left(\vec{x}-\vec{x}_p+\vec{x}_p\right)\times\left(\left(\vec{x}-\vec{x}_p+\vec{x}_p\right)\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv \tag{7.57}$$

$$\begin{aligned}\vec{A} = &\sum_p \int \vec{x}_p\times\left(\vec{x}_p\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv + \int \vec{x}_p\times\left(\left(\vec{x}-\vec{x}_p\right)\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv \\ &+ \int \left(\vec{x}-\vec{x}_p\right)\times\left(\vec{x}_p\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv + \int \left(\vec{x}-\vec{x}_p\right)\times\left(\left(\vec{x}-\vec{x}_p\right)\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv\end{aligned}$$

$$\begin{aligned}\vec{A} = &\sum_p \vec{x}_p\times\left(\vec{x}_p\times\vec{\alpha}_p\right)\int \zeta_\sigma\left(\vec{x}-\vec{x}_p\right)dv + \vec{x}_p\times\left(\left(\int\left(\vec{x}-\vec{x}_p\right)\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)dv\right)\times\vec{\alpha}_p\right) \\ &+ \left(\int\left(\vec{x}-\vec{x}_p\right)\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)dv\right)\times\left(\vec{x}_p\times\vec{\alpha}_p\right) + \int\left(\vec{x}-\vec{x}_p\right)\times\left(\left(\vec{x}-\vec{x}_p\right)\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv\end{aligned}$$

$$\vec{A} = \sum_p \vec{x}_p\times\left(\vec{x}_p\times\vec{\alpha}_p\right) + \int \left(\vec{x}-\vec{x}_p\right)\times\left(\left(\vec{x}-\vec{x}_p\right)\times\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right)dv \tag{7.58}$$

$$\vec{A} = \sum_p \vec{x}_p\times(\vec{x}_p\times\vec{\alpha}_p)+\int\left[\left(\vec{x}-\vec{x}_p\right).\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right]\left(\vec{x}-\vec{x}_p\right)dv-\int||\vec{x}-\vec{x}_p||^2\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)dv \tag{7.59}$$

$$\vec{A} = \sum_p \vec{x}_p\times(\vec{x}_p\times\vec{\alpha}_p)+\int\left[\left(\vec{x}-\vec{x}_p\right).\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right]\left(\vec{x}-\vec{x}_p\right)dv-4\pi\vec{\alpha}_p\int||\vec{x}-\vec{x}_p||^4\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)dr \tag{7.60}$$

$$\int\left[\left(\vec{x}-\vec{x}_p\right).\vec{\alpha}_p\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\right]\left(\vec{x}-\vec{x}_p\right)dv = \int||\vec{x}-\vec{x}_p||^2\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\left(||\vec{\alpha}_p||cos\phi\right)\vec{u}_{rp} \tag{7.61}$$

Where ϕ is the angle between $\vec{\alpha}_p$ and $\vec{x}-\vec{x}_p$. The last integral, which contains the term $cos\phi$, is symmetric with respect to $\vec{\alpha}_p$ and has a non null component in only $\vec{\alpha}_p$ direction. If we consider a reference frame centered at p and the z component is along $\vec{\alpha}_p$, the aformentiened integral reduces to :

$$\int||\vec{x}-\vec{x}_p||^2\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)\left[||\vec{\alpha}_p||cos\phi\right]\vec{u}_{rp} = \vec{\alpha}_p\int||\vec{x}-\vec{x}_p||^2\zeta_\sigma\left(\vec{x}-\vec{x}_p\right)cos^2\phi dv \tag{7.62}$$

$$\vec{\alpha}_p \int ||\vec{x}-\vec{x}_p|\,|^2 \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) r^2 dr \int cos^2\phi sin\phi d\phi \int_0^\pi d\theta = \frac{4\pi}{3}\vec{\alpha}_p \int ||\vec{x}-\vec{x}_p|\,|^2 \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dr \tag{7.63}$$

$$\vec{A} = \sum_p \vec{x}_p \times \left(\vec{x}_p \times \vec{\alpha}_p\right) - \frac{2}{3}\vec{\alpha}_p \int 4\pi\, ||\vec{x}-\vec{x}_p|\,|^4 \zeta_\sigma\left(\vec{x}-\vec{x}_p\right) dr \tag{7.64}$$

Let $C = 4\pi \int r^4 \zeta_\sigma(r) dr = 4\pi\sigma^2 \int \rho^4 \zeta(\rho) d\rho$ then we have:

$$\vec{A} = \sum_p \vec{x}_p \times \left(\vec{x}_p \times \vec{\alpha}_p\right) - \frac{2}{3} C\sigma^2 \sum_p \vec{\alpha}_p \tag{7.65}$$

$$\vec{A} = \sum_p \vec{x}_p \times \left(\vec{x}_p \times \vec{\alpha}_p\right) - \frac{2}{3} C\sigma^2 \vec{\Omega} \tag{7.66}$$

For any core function of third order and above we have $C = 0$ thus :

$$\vec{A} = \sum_p \vec{x}_p \times \left(\vec{x}_p \times \vec{\alpha}_p\right) \tag{7.67}$$

7.3 Appendix C: Variation of Total Circulation and Linear Impulse Upon Using Winckelmans Method to Resolve the Problem of the Vorticity Divergence

$$\begin{aligned}\vec{\Gamma} = \int \vec{\omega}_\sigma(\vec{x}) dv &= \sum_p \vec{\alpha}_p \int \left[\zeta_\sigma(\vec{x}-\vec{x}_p) - \frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p|\,|^3}\right] dv \\ &+ \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p|\,|^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right] \left(||\vec{\alpha}_p|\,|cos\phi\right) \vec{u}_{r_p} dv\end{aligned}$$

Where ϕ is the angle between $\vec{\alpha}_p$ and $\vec{x}-\vec{x}_p$. The last integral, which contains the term $cos\phi$, is symmetric with respect to $\vec{\alpha}_p$ and has a non null component in only $\vec{\alpha}_p$ direction. If we consider a reference frame centered at p and the z component is along $\vec{\alpha}_p$, the aformentiened integral reduces to :

$$\int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right](||\vec{\alpha}_p|\,cos\phi)\,\vec{u}_{r_p}dv = \vec{\alpha}_p \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right] cos^2\phi dv$$

$$\vec{\alpha}_p \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right] r^2 dr \int_0^\pi cos^2\phi sin\phi d\phi \int_0^{2\pi} d\theta =$$

$$\frac{4\pi}{3}\vec{\alpha}_p \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right] r^2 dr =$$

$$\frac{1}{3}\vec{\alpha}_p \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right] dv =$$

$$\vec{\alpha}_p \int \left[\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \frac{1}{3}\zeta_\sigma(\vec{x}-\vec{x}_p)\right] dv$$

Thus the total circulation can be expressed as:

$$\vec{\Gamma} = \sum_p \vec{\alpha}_p \int \left[\zeta_\sigma(\vec{x}-\vec{x}_p) - \frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3}\right] + \left[\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \frac{1}{3}\zeta_\sigma(\vec{x}-\vec{x}_p)\right] dv \tag{7.68}$$

$$\vec{\Gamma} = \sum_p \frac{2}{3}\vec{\alpha}_p \int \zeta_\sigma(\vec{x}-\vec{x}_p)dv \tag{7.69}$$

$$\vec{\Gamma} = \frac{2}{3}\sum_p \vec{\alpha}_p \tag{7.70}$$

$$\vec{I} = \int \vec{x}\times\vec{\omega}_\sigma(\vec{x})dv = \sum_p -\vec{\alpha}_p \times \int \left[\zeta_\sigma(\vec{x}-\vec{x}_p) - \frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3}\right]\vec{x}dv$$
$$+ \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right](||\vec{\alpha}_p|\,cos\phi)\,(\vec{x}\times\vec{u}_{r_p})dv$$

$$\vec{I} = \sum_p \vec{x}_p \times \vec{\alpha}_p \int \left[\zeta_\sigma(\vec{x}-\vec{x}_p) - \frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3}\right] dv$$
$$- \vec{\alpha}_p \times \int \left[\zeta_\sigma(\vec{x}-\vec{x}_p) - \frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3}\right](\vec{x}-\vec{x}_p)dv$$
$$+ \vec{x}_p \times \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right](||\vec{\alpha}_p|\,cos\phi)\,\vec{u}_{r_p}dv$$
$$+ \int \left[3\frac{q_\sigma(\vec{x}-\vec{x}_p)}{||\vec{x}-\vec{x}_p||^3} - \zeta_\sigma(\vec{x}-\vec{x}_p)\right](||\vec{\alpha}_p|\,cos\phi)\,(\vec{x}-\vec{x}_p)\times\vec{u}_{r_p}dv$$

The last integral is zero since $\vec{x}-\vec{x}_p$ is in the same direction of $\vec{u}_{r_p}$. The second integral is also equal to zero since it is symmetric with respect to the source location p. Noting that we have an expression for the third integral, the linear impulse is expressed as :

$$\vec{I} = \sum_p \vec{x}_p \times \vec{\alpha}_p \int \left[\zeta_\sigma(\vec{x} - \vec{x}_p) - \frac{q_\sigma(\vec{x} - \vec{x}_p)}{||\vec{x} - \vec{x}_p|\,|^3} \right] dv$$
$$+ \vec{x}_p \times \vec{\alpha}_p \int \left[\frac{q_\sigma(\vec{x} - \vec{x}_p)}{||\vec{x} - \vec{x}_p|\,|^3} - \frac{1}{3}\zeta_\sigma(\vec{x} - \vec{x}_p) \right] dv$$

$$\vec{I} = \frac{2}{3} \sum_p \vec{x}_p \times \vec{\alpha}_p \int \zeta_\sigma(\vec{x} - \vec{x}_p) dv \tag{7.71}$$

$$\vec{I} = \frac{2}{3} \sum_p \vec{x}_p \times \vec{\alpha}_p \tag{7.72}$$

7.4 Appendix D: Extra Results

Figures 7.2, 7.4, and 7.6 show the time development of contours and vorticity lines, as seen from the z direction for the cases $Re = 1000$, 1500, and 2000, respectively. The corresponding iso-vorticity surfaces, as seen from the x and y directions are shown in Figs. 7.3, 7.5, and 7.7.

Figures 7.8 and 7.9 show the L_2 norm for the truncation and regularization error versus p using a second order algebraic core function and second order Gaussian core function. Figures 7.10 and 7.11 show the L_2 norm of the truncation and regularization errors versus d for a fixed order of expansion $p = 5$ using a second order algebraic core function and second order Gaussian core function.

Figure 7.12 show the measured simulation time versus the number of elements for different n_F using an order of expansion $p = 8$.

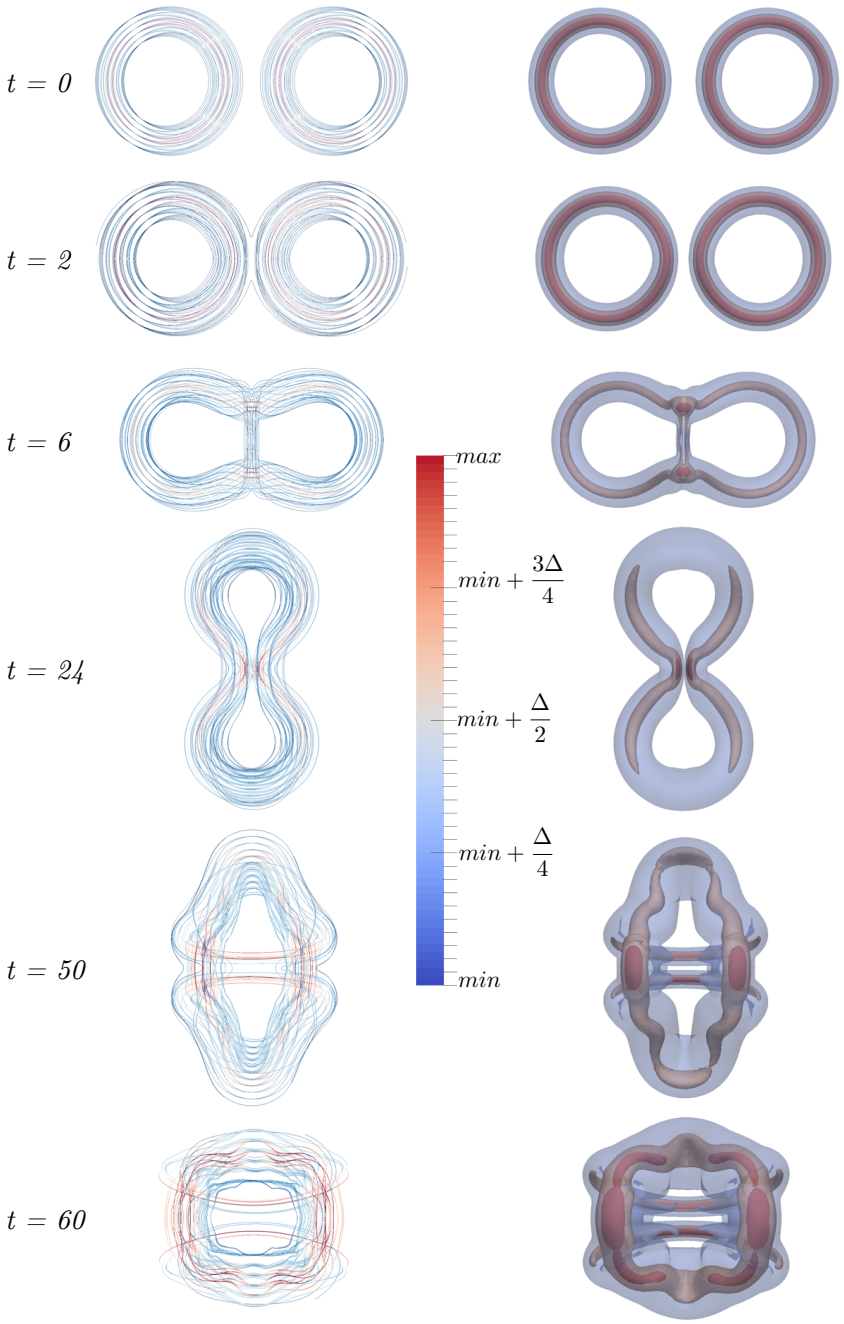

Figure 7.2: Iso -surfaces of vorticity norm and a random collection of vorticity lines at several representative stages of evolution for $Re = 1000$ seen from z direction. The levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm

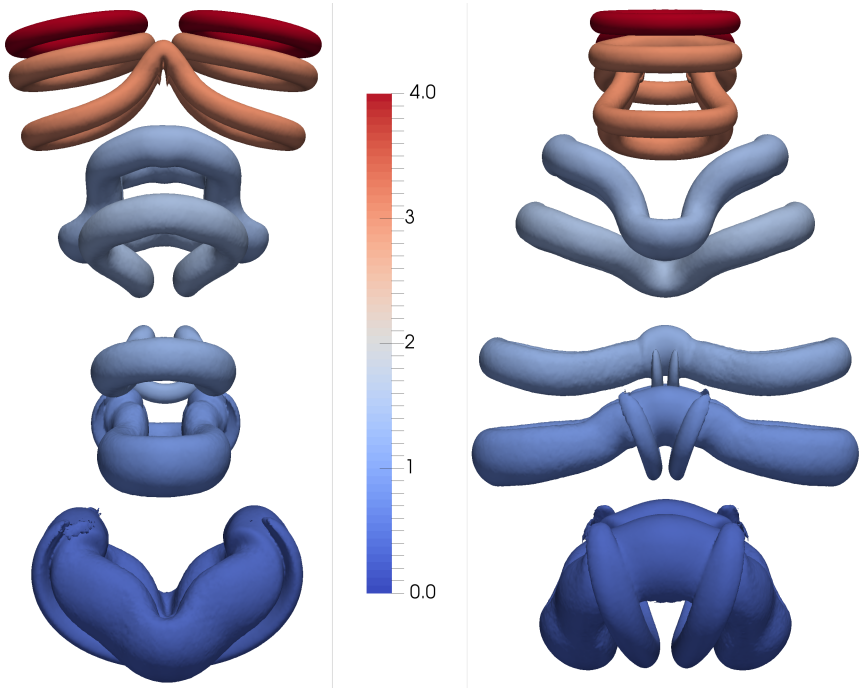

Figure 7.3: Iso -surfaces of vorticity norm at several representative stages of evolution for $Re = 1000$ seen from x and y directions. The level of the iso-surface is 25% of the instantaneous maximum of the vorticity norm

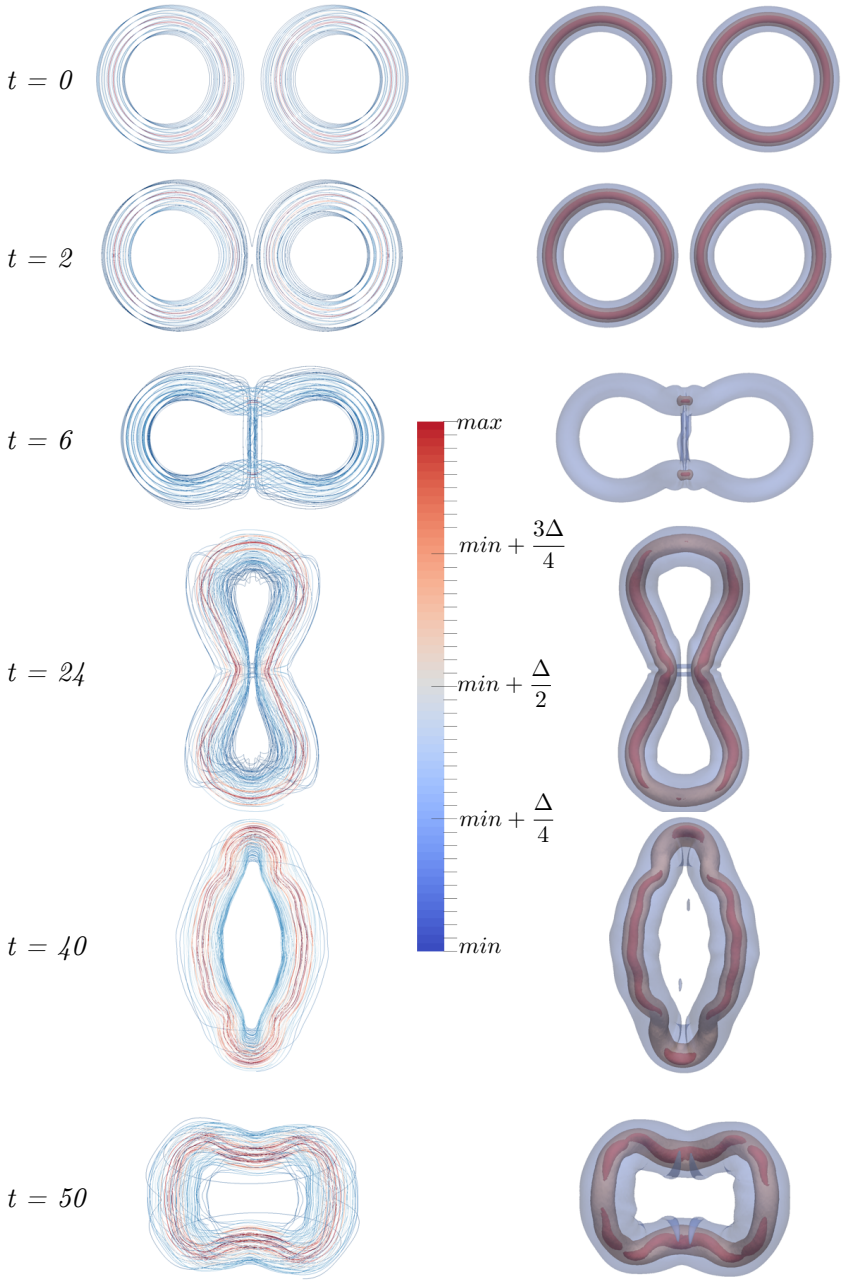

Figure 7.4: Iso -surfaces of vorticity norm and a random collection of vorticity lines at several representative stages of evolution for $Re = 1500$ seen from z direction. The levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm

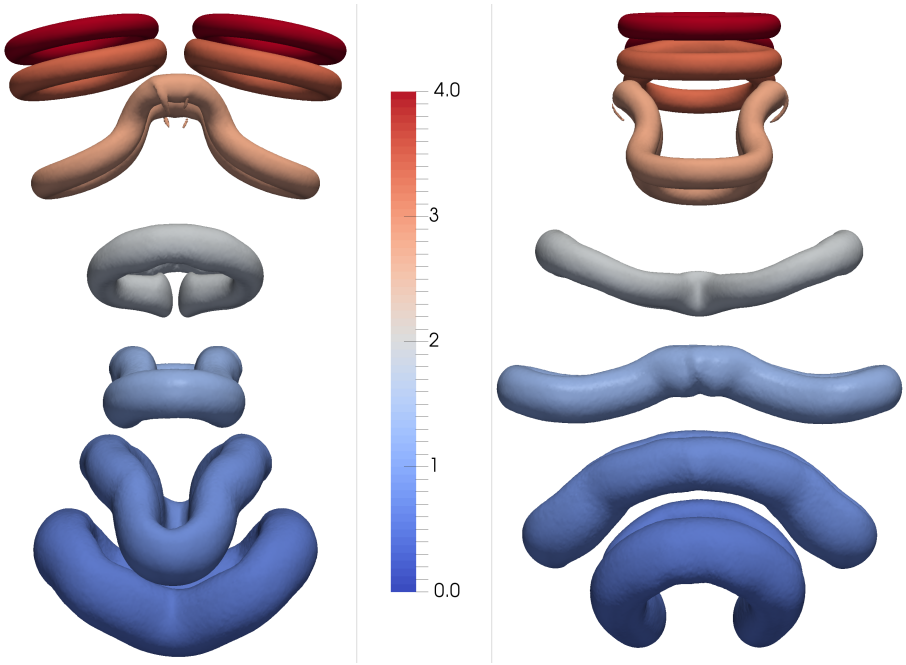

Figure 7.5: Iso -surfaces of vorticity norm at several representative stages of evolution for $Re = 1500$ seen from x and y directions. The level of the iso-surface is 25% of the instantaneous maximum of the vorticity norm

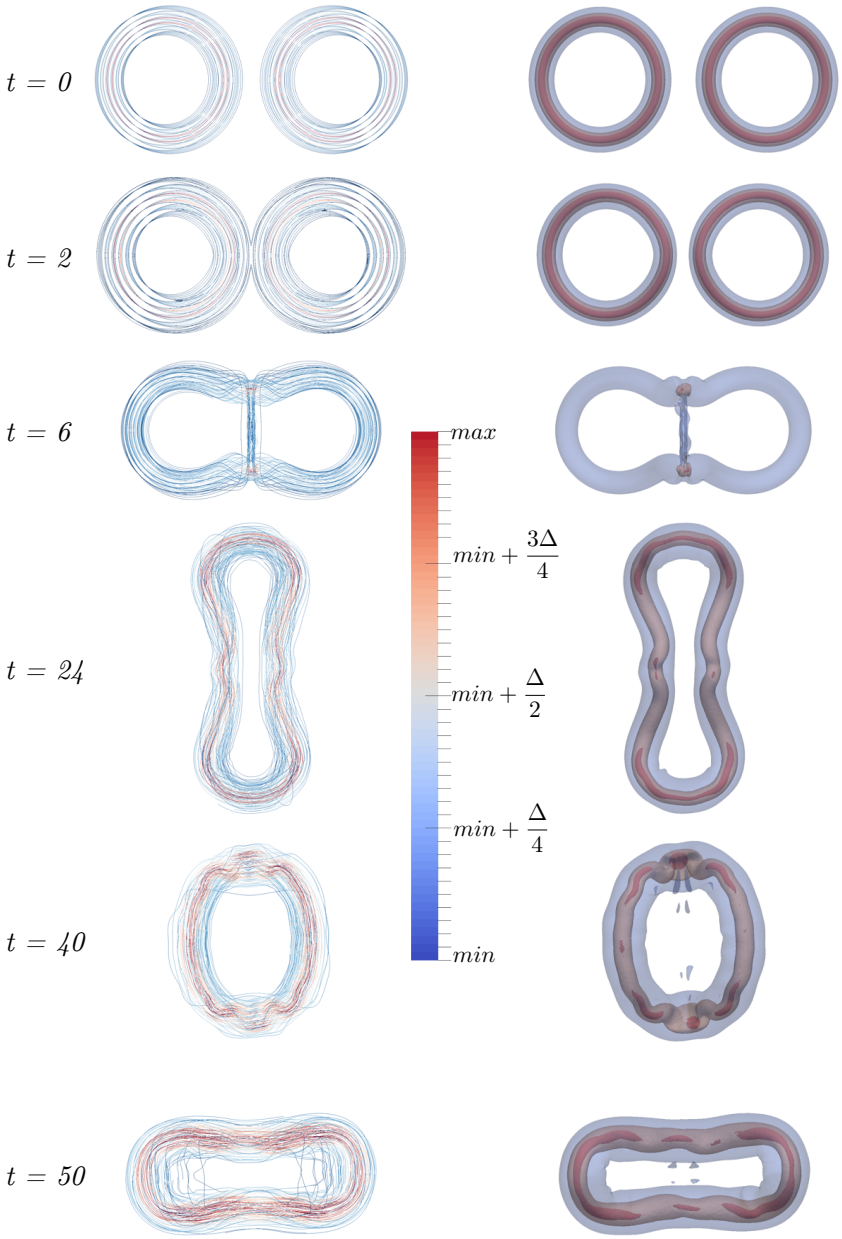

Figure 7.6: Iso -surfaces of vorticity norm and a random collection of vorticity lines at several representative stages of evolution for $Re = 2000$ seen from z direction. The levels of the iso-surfaces are 10%, 50%, and 80% of the instantaneous maximum of the vorticity norm

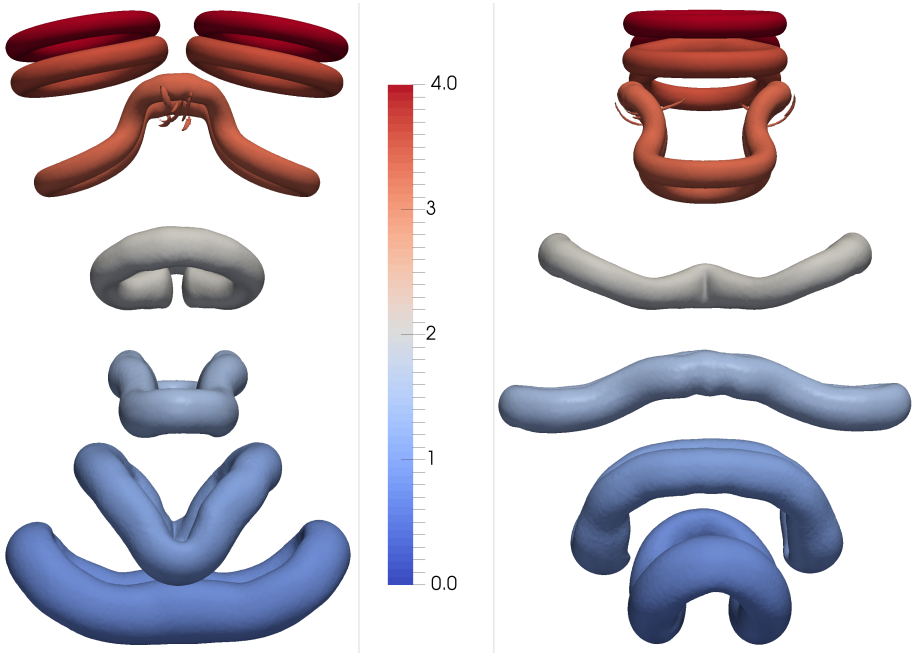

Figure 7.7: Iso -surfaces of vorticity norm at several representative stages of evolution for $Re = 2000$ seen from x and y directions. The level of the iso-surface is 25% of the instantaneous maximum of the vorticity norm

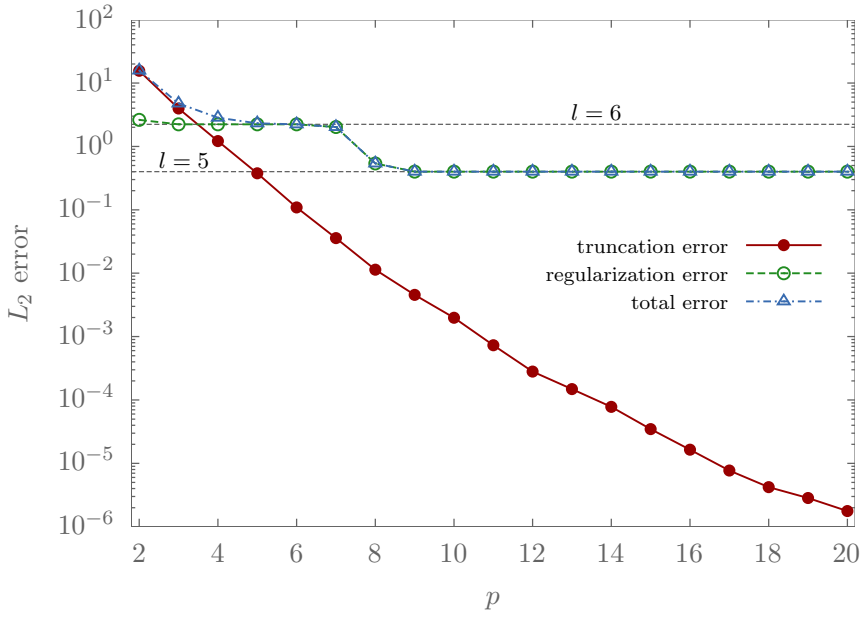

Figure 7.8: L_2 norm for the truncation and regularization error versus p using a second order algebraic core function in the context of adaptive scheme.

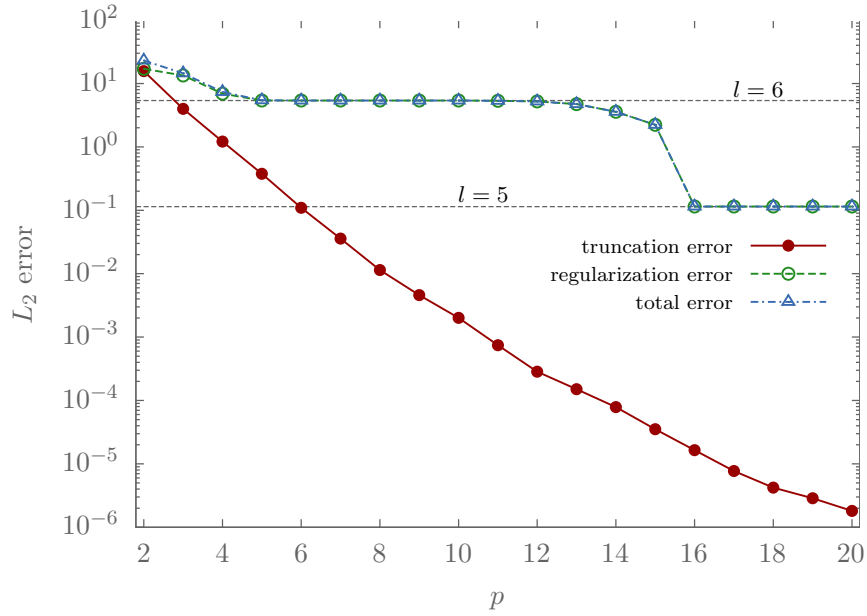

Figure 7.9: L_2 norm for the truncation and regularization error versus p using a second order Gaussian core function in the context of adaptive scheme.

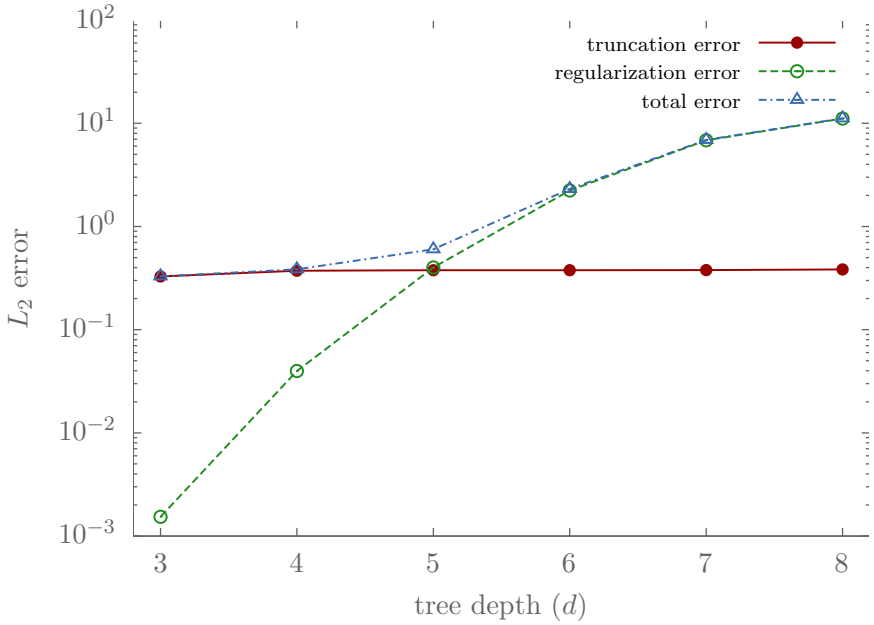

Figure 7.10: L_2 norm of the truncation and regularization errors versus d_σ for a fixed order of expansion $p = 5$ using a second order algebraic core function in the adaptive scheme.

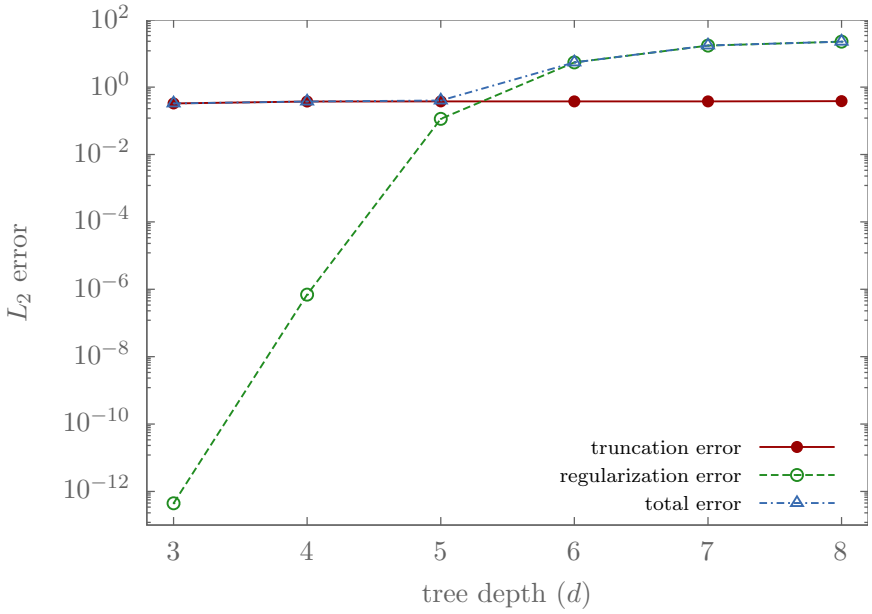

Figure 7.11: L_2 norm of the truncation and regularization errors versus d_σ for a fixed order of expansion $p = 5$ using a second order Gaussian core function in the adaptive scheme.

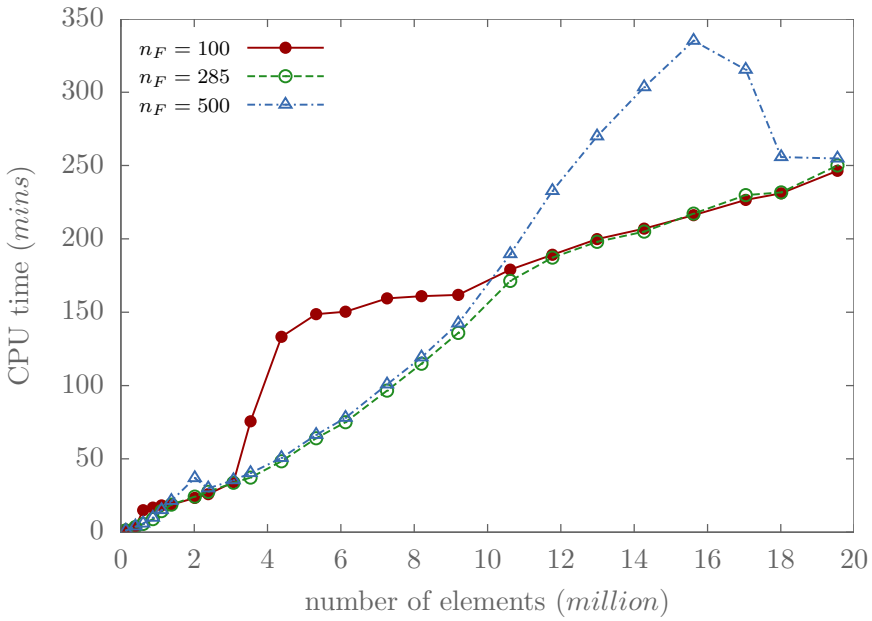

Figure 7.12: Measured simulation time versus the number of elements for different n_F using an order of expansion $p = 8$.