Three-Dimensional Vortex Particle-In-Cell Technique

Damien Cox

Department of Mechanical & Manufacturing Engineering,
Trinity College,
Dublin 2,
Ireland.

October, 2004

A thesis submitted to the University of Dublin in partial fulfilment of the requirements for the degree of Ph.D.

Abstract

A three-dimensional vortex particle-in-cell technique is presented. This technique is applied to the simulation of unsteady external flows. While suited to flows where convection effects dominate, a wide range of Reynolds numbers can be simulated, with diffusion solely due to numerical error at the upper limit.

Velocity calculations are performed on a regular Eulerian grid using an FFT solver for the Poisson equation in velocity-vorticity formulation. Standard high-order interpolation functions are used, both for transferring information between the mesh and the particles, and for performing regular redistribution of the particles. The viscous diffusion term is solved directly on the Eulerian mesh.

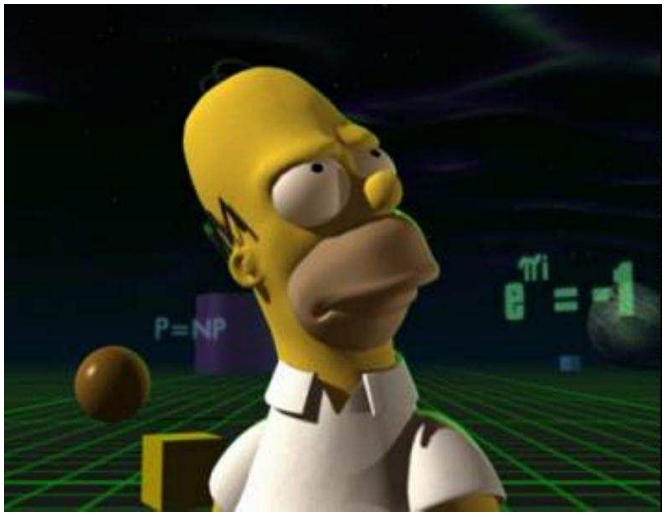
Results are presented for azimuthally perturbed vortex rings with a range of perturbation wavenumbers. This is a 'quasi-Eulerian' simulation. It is shown that certain wavenumbers cause the ring to become unstable, as predicted by stability theory. Impulsively started jets are also modelled. The circulation shedding rate at the jet lip is governed by a set of linear equations, which are solved at each shedding interval.

Declaration

I declare that I am the author of this thesis and that all work described herein is my own, unless otherwise referenced. Furthermore this work has not been submitted in whole or part, to any other university or college for any degree or qualification.

I authorise the library of Trinity College, Dublin to lend this thesis.

Damien Cox, October 2004



LISA Well, where's my Dad?

FRINK Well, it should be obvious to even the most dim-witted individual who holds an advanced degree in hyperbolic topology, n-heh, that Homer Simpson has stumbled into...

The lights go off.

FRINK ...the third dimension.

Lisa turns the lights back on.

LISA Sorry.

FRINK Here is an ordinary square...

WIGGUM Whoa, whoa, slow down, Egghead!

FRINK ...but suppose we extend the square beyond the two dimensions of our universe along the hypothetical "z-axis", there.

Frink draws a wireframe cube on the blackboard.

EVERYONE *Gasps.*

FRINK This forms a three-dimensional object known as a "cube", or a "Frinkahedron" in honour of its discoverer, n-huh, n-hey...

Contents

List of Figures

List of Tables

Nomenclature

$\mathbf{x}_p$	Particle position vector
$\boldsymbol{\alpha}_p$	Particle strength vector
$\boldsymbol{\omega}$	Vorticity vector
$\mathbf{u}$	Velocity vector
Γ	Circulation
Ψ	Stream function
$\nabla\mathbf{u}$	Velocity gradient tensor
$\mathbb{S}$	Rate-of-strain tensor
$\mathbb{R}$	Rotation tensor
$\mathbf{u}-\boldsymbol{\omega}$	Velocity-vorticity formulation
$\Psi-\boldsymbol{\omega}$	Stream function-vorticity formulation
VIC	**V**ortex **I**n **C**ell
RHS	**R**ight **H**and **S**ide
BC	**B**oundary **C**ondition
FFT	**F**ast **F**ourier **T**ransform
CFD	**C**omputational **F**luid **D**ynamics

Chapter 1

Introduction

Computational fluid dynamics (CFD) has fast evolved from a mathematical curiosity to a standard tool in the analysis of fluid flows. It is defined as the numerical solution, by computational methods, of the governing equations which describe fluid flows. The attraction of CFD is that it offers a convenient means of understanding flows which can prove either too difficult or overly expensive for traditional experimental techniques. The sharp increase in computational power since CFD was first introduced has presented more and more opportunities to model flows previously analysed only in experiments.

However, this computational power has a finite upper limit, and so there will always be a place in CFD research for investigating novel techniques for solution of the governing equations, in the hope that they will offer improved performance over traditional approaches. *Vortex methods* are a family of techniques that fall into this category, and they have been investigated extensively since their first use by Rosenhead [5].

The class of vortex methods examined in the current study are *Particle-In-Cell* techniques. Basically, these techniques share a common feature with all vortex methods in that it is the development of the vorticity field which is examined, and this vorticity field is discretised onto a set of particles. These particles are then convected in a velocity field which is calculated on a fixed mesh. This hybrid

approach 'cherry-picks' the best features of the Lagrangian and the Eulerian approach, to give a combined algorithm more efficiency and robust than the two in isolation. This will be discussed in more detail in chapter 2.

A common feature in all of the flows analysed herein is the presence of coherent vortex structures. Examples of such structures in engineering are the vortex ring structures in the potential core region of free jets, as illustrated in Fig. 1.1, or the trailing vortices from the wingtips of aircraft.

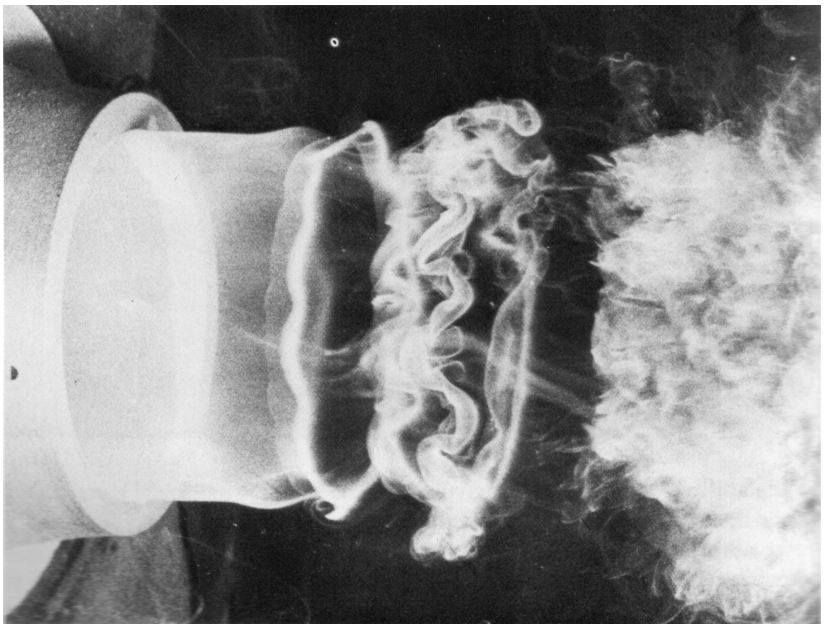


Figure 1.1: Vortex ring instability in a round jet at a Reynolds number of about 13,000

Geese flying in v-formation make use of shed vortices to reduce lifting effort, allowing them to cover distances of up to 3,000 miles during migration. Certain insects make complex use of vortex shedding, in a way only partially understood, to produce lift up to three times greater than believed possible using scaled-down aerodynamic models. An interesting new study has also used vortices to explain the locomotion of the "water boatman", a long-legged insect that relies on surface tension on the water's surface to stay afloat. The backward momentum of vortices

generated by motion of its legs propels the insect forward. This motion is clearly illustrated in Fig. 1.2.

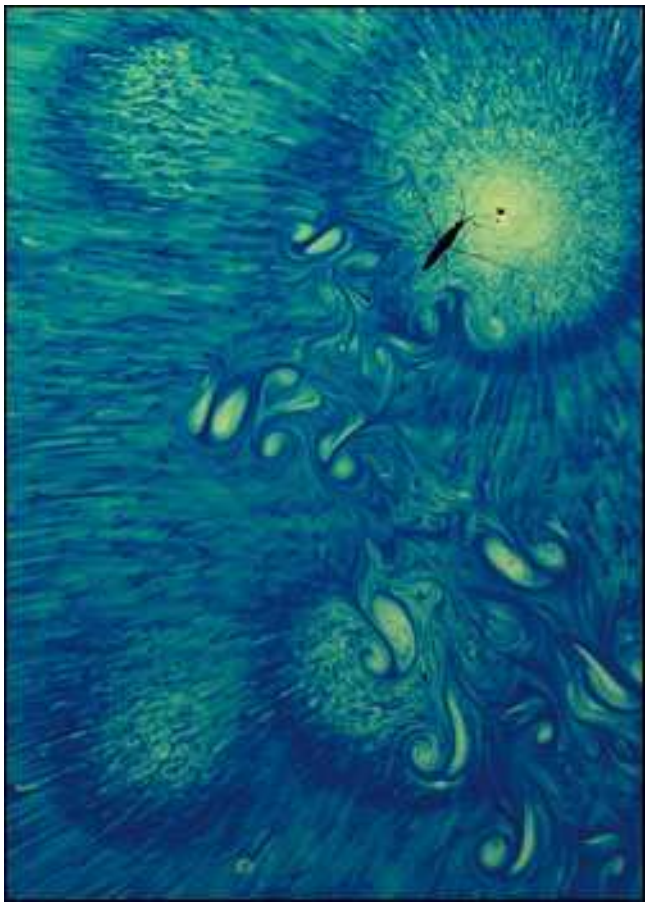


Figure 1.2: U-shaped vortices formed by water boatman locomotion

While the modelling of jets similar to that shown in Fig. 1.1 is the primary aim of this study, *vortex rings* are used extensively in this work to test various aspects of the model. Generation of these structures is most familiar as a measure of a smoker's prowess, though similar structures have been witnessed in nature from the craters of volcanoes. Measuring up to 200m in diameter, a very clear example is shown in Fig. 1.3.

The modelling of jets like that shown in Fig. 1.1 is important for several reasons. Noise reduction from commercial aircraft is a huge area of research at present, as strict new environmental noise legislation requires drastic reductions

Figure 1.3: Vortex 'steam' ring shed from Mount Etna, Sicily, on 24th February 2000

in all aspects of aircraft noise. On takeoff, jet noise is one of two dominant noise sources (the other is fan noise) and can generate noise levels over 90dB. Attempts to reduce this noise usually consist of modifying the shear layer structure in some way, either by shielding the central core with a lower speed jet (co-axial configuration) or by using modified exhaust nozzle shapes to break up the strong velocity gradients (*e.g.* scalloping). Modelling the effect of these modifications on the flow field is important to understand their effect on the resulting noise generated.

There are three main categories of CFD which have been applied to the problem of jet flows. These are, in order of computational power required: **R**eynolds-averaged **N**avier-**S**tokes, **L**arge-**E**ddy **S**imulation and **D**irect **N**umerical **S**imulation.

In the RANS formulation, only the mean flow is resolved. Both small and large scales are time-averaged, and their effects are modelled via the turbulent Reynolds stress. Various turbulence models have been developed for use with this technique, the standard being $k-\epsilon$ though with ASM (Algebraic Reynolds stress

modelling) being used increasingly frequently in the literature. The applicability of RANS is limited however, as no transient data is generated. A more recent development, **U**nsteady **RANS**, offers a means of recovering at least part of the turbulent data, as the unsteady term is retained in the usual RANS equations. However, a turbulence model is still required in the spatial discretisation, so it cannot fully resolve the turbulent data.

In LES only the large scales of the flow must be resolved, and a *sub-grid scale* Reynolds stress is used to model their influence on the smaller, unresolved, scales. This increases greatly the Reynolds number which can be modelled, though large parallel computers are still required to perform calculations. Jets with Reynolds numbers of the order of $Re_D = 1.0 \times 10^5$ have been successfully modelling using this approach [6].

In DNS, all scales of the turbulent flow must be resolved, which requires an extremely fine mesh, and therefore huge computational power, even for jets with Reynolds number of a few thousand.

A recently developed hybrid technique uses RANS at certain mesh and immersed boundaries and LES for the free field dynamics. This greatly reduces computational effort required, especially for complex geometries, such as scalloped nozzles.

Understanding the fluid dynamics of jet flows is usually focussed on the behaviour of coherent vortex structures in the flow, *e.g.* the reorientation from azimuthal to streamwise vorticity and the ultimate breakdown of these structures to fully turbulent flow. Modifications to the nozzle geometry has a direct impact on these features, for example scalloping of the jet nozzle excites the streamwise vorticity generation mechanism and accelerates the vortex breakdown. It is therefore an intuitive approach to use vortex elements as a basis in the modelling of such flows, as it is their behaviour that is of the most interest.

These vortex elements are discrete packets of vorticity, and it will be shown in later sections that by using these elements, or 'particles', it is only necessary

to track their change in position and strength due to local gradients in velocity and vorticity in order to adequately represent the entire flow field.

1.1 Objectives & Outline

As a first step to solving a flow from a full-scale jet engine, the objective of this work is to develop a computationally inexpensive code, based on vortex methods. Parallel computing strategies are not considered, as the focus is on the underlying algorithm development. This requires a code that can solve for fully three-dimensional transient incompressible viscous flow on a single computer processor. A suite of algorithms is developed to deal with each aspect of the discretisation of the governing equations.

Work carried out in the field of vortex methods, relevant to the current study, is summarised in **Chapter 2**. Some basic concepts of fluid dynamics and vortex methods are briefly outlined in **Chapter 3**. The development of each algorithm in the solver is discussed in detail in **Chapter 4**. Sample applications of the solver are described in **Chapter 5**, and the results of these applications are presented in **Chapter 6**.

Chapter 2

Literature Review

2.1 Vortex Methods

Vortex methods for the study of fluid flows were initiated by L. Rosenhead in 1931 [5]. He used a small system of singular point vortices to represent an inviscid shear layer. When the particle locations were perturbed sinusoidally, the sheet rolled up, demontrating the Kelvin-Helmholtz instability. This behaviour is shown in the series of images, Fig. 2.1.

Rosenhead was only able to solve four time iterations however, as he had to perform all calculations by hand.

Subsequent work by other authors attempted to improve the accuracy of the same technique by performing their calculations on early computers, but quickly

ran into trouble with very large induced velocities when two point vortices approached too close to each other, due to the singular nature of their vorticity distribution.

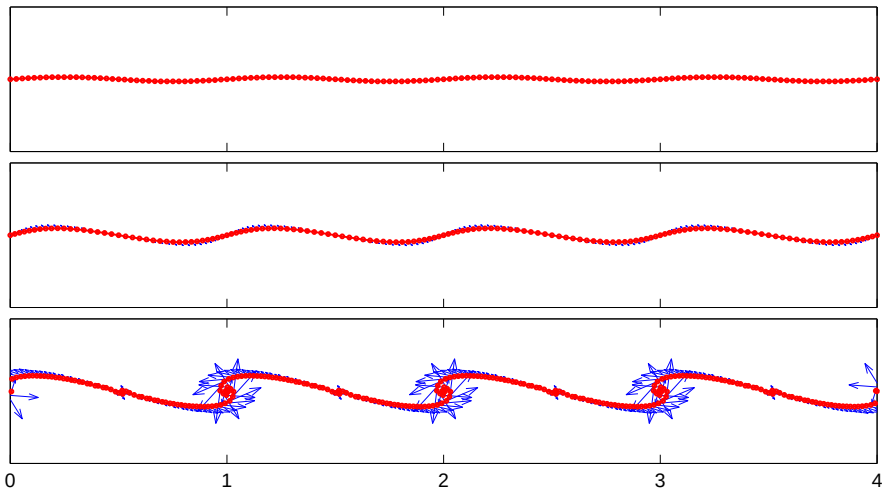


Figure 2.1: Shear layer instability as demonstrated by Rosenhead

Vortex blobs, introduced by Chorin in [7], remove the singularity, either by directly modifying the velocity field or by replacing the Dirac delta function with a smooth vorticity distribution model. Chorin used the first approach and applied a cutoff function that limited the maximum possible velocity. Modern discrete vortex methods use the second approach; a wide variety of core models have been designed, (see [3] for an exhaustive list). Recent work has achieved excellent spatial resolution with this method, provided certain convergence criteria are observed [8]. The velocity fields induced by a singular particle, a Chorin-type particle, and a standard Gaussian core model vortex are plotted in Fig. 2.2.

This singularity issue is the first of two fundamental problems which must be overcome when developing a vortex method. The second is a problem common to all multi-body calculations. In order to calculate the interaction between a group of N particles, $\mathcal{O}(N^2)$ calculations are required. On modern computers, this computational cost quickly becomes prohibitive for any value of N greater than a few hundred.

Several techniques have been developed to reduce this cost. Purely Lagrangian approaches subdivide into two closely related techniques, *treecode* algorithms, [9], and *fast multipole* methods, [10]. The basic concept behind both of these techniques is the same–to subdivide the particle field into a hierarchy of clusters based on their spatial locations. Depending on two clusters' separation, their effects on each other can be treated as single cluster-cluster interactions, or particle-cluster interactions.

In a treecode algorithm, the particles are divided into a nested set of groups and the $\mathcal{O}(N^2)$ particleparticle interactions are replaced by a smaller number of particlecluster interactions. The fast multipole method is more complex, requiring the use of spherical harmonics in three dimensions. The FMM also employs a more elaborate evaluation procedure in which the far-field multipole approximation is modified at the local particle level. Both of these algorithms can lead to computations that scale linearly with the number of particles, $\mathcal{O}(N)$.

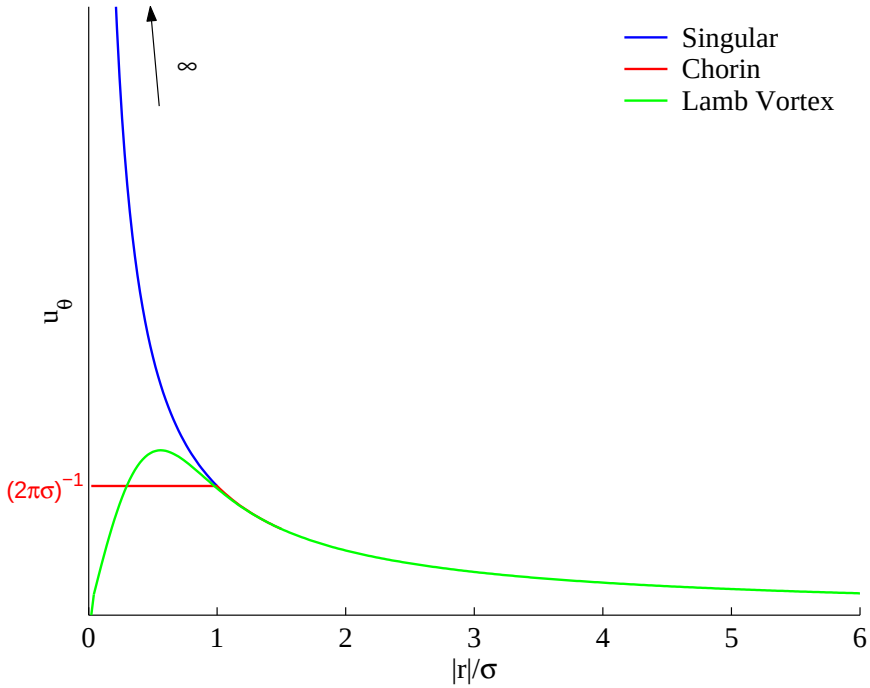


Figure 2.2: Velocities induced by singular and regularised particles

The fast multipole method has been implemented successfully in 2D for computations involving millions of particles [8]. The treecode algorithm outperforms the fast multipole method in three-dimensional simulations, due to its simpler hierarchy interactions. An excellent example of its application is given in [11].

In the same year as Chorin's seminal paper, Christiansen [12] sidestepped the $\mathcal{O}(N^2)$ nature of the velocity calculation, while also avoiding the singularity problem in a different way. He applied the recently developed *particle-in-cell* [PIC] [13] technique to fluid-flow problems using vortex methods.

Maintaining a singular vorticity distribution on the vortex particles, a fixed Eulerian grid is used to calculate their induced velocity field. Clearly, there can be no singular velocities as there is an implicit smoothing involved in reconstructing the vorticity field on the underlying grid. Fast Poisson solvers can be used for the velocity calculations, yielding computational performance of $\mathcal{O}(M \log M)$, where M is the number of grid points. This technique is described in detail in § 2.2, as it forms the basis of the work in this thesis.

As for the question of PIC efficiency versus pure Lagrangian techniques, this cannot be answered simply. The efficiency of both approaches is entirely dependent on the geometry of the problem being modelled. For example, a purely Lagrangian approach will always outperform a PIC approach for modelling wingtip vortices or other highly compact elongated region of vorticity. For regions of distributed vorticity, the PIC approach is much faster. This is summarised mathematically in [11] where the performance of both approaches is compared for different particle densities in the same domain.

Computational power required for several different computations in the literature are summarised in Fig. 2.3.

2.1.1 Viscous Diffusion

All real fluids have some degree of resistance to shearing forces. This is the *viscosity*. Modelling of viscosity is an important requirement for a practical vortex

method. Early techniques primarily modelled inviscid flows, as do most current vortex filament techniques. However, particle-based vortex methods offer a range of ways of accurately incorporating viscous effects into a simulation.

Viscosity tends to decrease gradients in the vorticity field, smoothing out peaks and increasing the overall support of the vorticity. While redistribution of particle strengths is the obvious approach to modelling this effect, altering particle locations can have a similiar effect, as seen in the random walk method.

Three separate approaches were examined in this study.

Random Walk Simplest means of modelling diffusion, requires large number of particles, unsuitable for 3D.

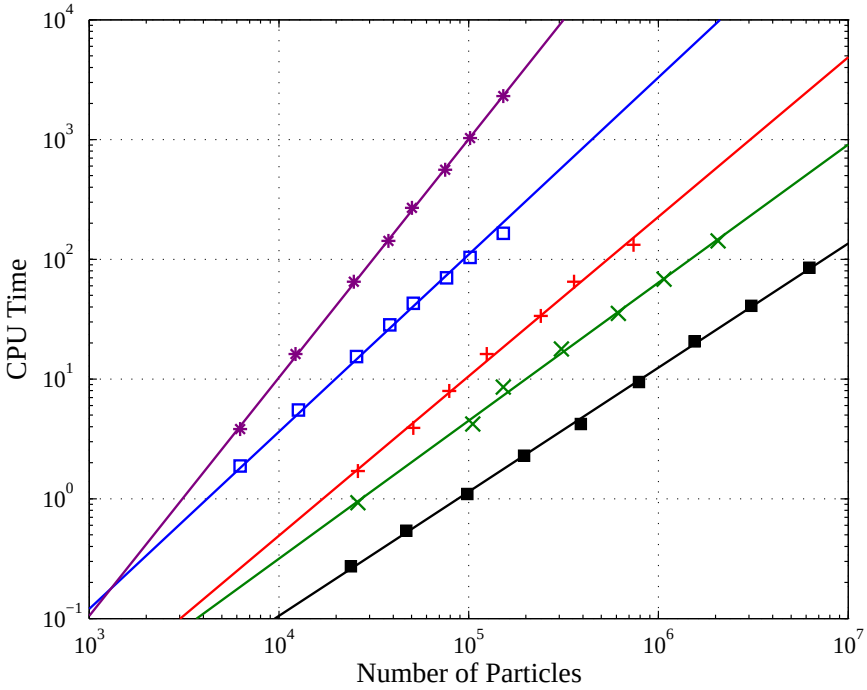


Figure 2.3: Computational effort for velocity calculations vs. number of particles, (*) Direct summation, (□) Fast multipole method, (+) VIC method on a cylindrical grid filled with 25% particles, (×) VIC method on a cylindrical grid filled with 65% particles, (■) VIC method on a cartesian grid filled with 100% particles

Particle Strength Exchange Offers highest resolution, very computationally expensive

Mesh-based Limited resolution, very efficient.

Chorin was the first to use the *random vortex method*. This method is based on the fact that the probability of existence of a particle moving at random, like Brownian motion, is determined by the diffusion equation. The random walk added to the motion of the vortices reproduces the viscous diffusion in a statistical sense. While this is a very attractive technique for its simplicity it does have its limitations. A huge number of particles is required to ensure accurate representation of the diffusion mechanism, which caused problems for early researchers using this technique. Chorin himself wrote "we do not expect valid solutions at low Re", due to the coarseness of the solution when the magnitude of the random walk becomes comparable to the convective effect. More recent work, for example [14], has yielded very satisfactory results using this technique. In the cited work, $\mathcal{O}(10^5)$ particles were used to model 2D flow over a cylinder in cross flow.

Unfortunately, applying a random walk to three-dimensional particles violates Helmholtz's law. This introduces a random divergence into the vorticity field, so while its efficiency wold be even more appreciated in a three-dimensional calculation, it is unsuitable for this application.

The *Particle Strength Exchange* technique is a newer, more advanced, algorithm. Developed in [15], it has largely superceded all other purely Lagrangian techniques. The concept of this technique is to replace the Laplacian diffusion operator by an integral one,

$$\nabla^2 \boldsymbol{\omega} = \sigma^2 \sum_q (\mathrm{vol}_p \boldsymbol{\omega}_q^h - \mathrm{vol}_q \boldsymbol{\omega}_p^h) \eta_\sigma (\mathbf{x}_q^h - \mathbf{x}_p^h) \tag{2.1}$$

Where η_σ is the core model function. While each particle only affects a small number of other particles surrounding it (η is guaranteed to have compact support), the computational cost of this technique is still $\mathcal{O}(N^2)$, thus requiring some

form of multipole expansion for efficient solution.

The nature of the technique also places strong restrictions on particle core overlapping, usually required frequent remeshing to ensure this criteria is met. This can cause difficulty for problems involving immersed boundaries.

The third technique examined is made possible by the use of a fixed Eulerian mesh in the particle-in-cell approach. The diffusion equation can be solved directly on the mesh, and the results can be spread onto the particles using the same interpolation functions as for the vorticity and velocity. This treatment of viscous diffusion is discussed in greater detail in §.4.3.

2.1.2 Three Dimensions

Extension of vortex methods to three dimensions is not trivial. The stretching term, $\boldsymbol{\omega} \cdot \nabla \mathbf{u}$ in the vorticity transport equation Eq. 3.21 no longer reduces to zero, as is the case in two-dimensional flow, and must be modelled with very high accuracy, as it is now the dominant mechanism responsible for redistribution of the vorticity field. It is the treatment of this term that subdivides the two different approaches to this problem.

Vortex *filaments*, introduced by Leonard, [16, 17, 18], directly apply Helmholtz's law that vortex structures must move as material lines. All that is necessary to track the development of these structures is to track the movement of the representative material lines. The stretching term is accounted for by ensuring that the vorticity vectors lie everywhere tangent to the material lines.

There are two main drawbacks to use of this technique. For viscous flows, the vortex filaments no longer retain their identity. In fact there is no guarantee that viscosity will alter the circulation uniformly around the circumference of a vortex filament, and so the technique is invalidated. A further problem arises in the generation of vortex filaments from immersed boundaries, where the exact structure of the filaments shed from the surface must be known *a priori*. These factors greatly limit the technique's applicability beyond idealised inviscid flows.

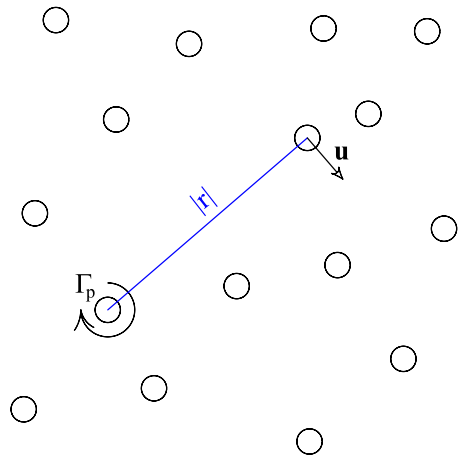
Vortex *particles*, or *vortons*, or *vortex sticks* as they have been variously termed, were first used by Rehbach, [19, 20] for modelling three-dimensional vortex sheets. They are a direct extension of the classical two-dimensional vortex particle scheme, with the necessary additions to handle vortex stretching. Details of the treatment of this term are in §.4.2. The vortex particle technique has largely superseded vortex filaments, as models can incorporate both viscous diffusion and complex immersed boundaries.

2.2 Particle In Cell

The concept of classical particle-in-cell in the context of vortex methods is explained here. Current developments involve higher order spatial interpolation and temporal integration, but the fundamental principal of the method has not changed.

The basic problem required to be solved by discrete vortex techniques is sketched in Fig. 2.4. A distributed group on particles, of number N, each carrying circulation are used to represent a smooth vorticity field. The induced velocity at each particle location is calculated by summing the influence of every other particle in the field, so the number of calculations required is $\mathcal{O}(N^2)$. Clearly, computational power required becomes prohibitively expensive for even a modest number of particles.

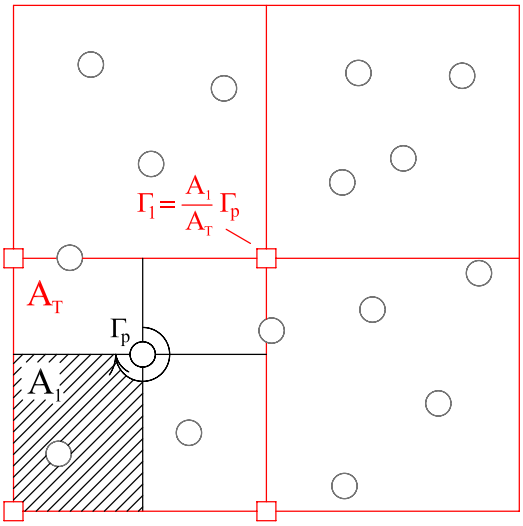
A further disadvantage of the direct Lagrangian approach is the requirement of a vortex core model to avoid singular velocities if two particles approach too closely. Use of such a model places stringent particle overlap restraints on this approach to ensure solution convergence. Also, the choice of function and support of the core model is quite arbitrary, and has no real physical grounding.



- $\mathcal{O}(N^2)$ calculations
- Core model required

Figure 2.4: Direct calculation of induced velocity

The original vortex Particle-in-Cell technique as developed by Christiansen [12] is designed specifically to overcome these two drawbacks. The vortex particle distribution is overlaid with a fixed Eulerian grid. Using some form of particle-mesh interpolation–in this instance a simple area-weighting–the particle circulations are spread onto the nodes of the fixed grid. The vorticity field can now be calculated by scaling the nodal circulations by the areas associated with each grid node. This step is illustrated in Fig. 2.5.



- Spread circulations Γ_p to mesh
- Simple area weighting
- Scale by area to give $\boldsymbol{\omega}_m$

Figure 2.5: Step 1 of PIC algorithm – Spread Γ_p

Once the vorticity field has been reconstructed on the grid, a Poisson equation is solved to calculate the velocities at the mesh nodes. Christiansen uses a stream function-vorticity formulation ($\Psi - \boldsymbol{\omega}$),

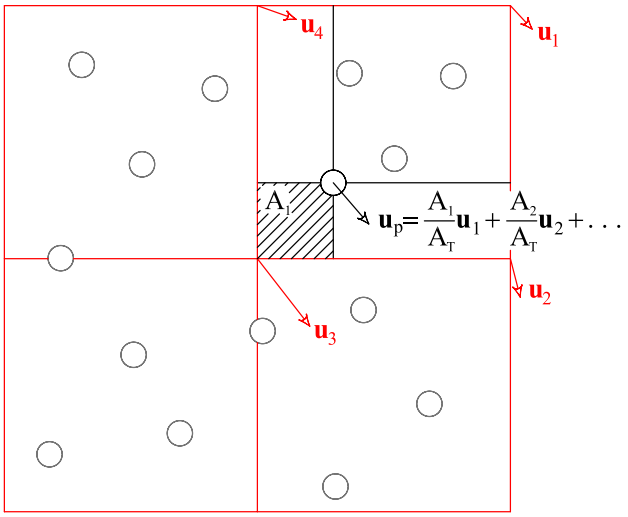
$$\nabla^2 \Psi = -\boldsymbol{\omega}$$

An advantage of this approach is that it is now only necessary to solve a single Poisson equation (In 3D this advantage does not apply). The two velocity components are calculated by differentiation of the stream function on the grid. However, this clearly reduces the order of continuity of the solution by one, which will limit the spatial resolution of the solution, making it necessary to have a high mesh density in locations of high strain.

An alternative approach to the stream function formulation is to use a velocity-vorticity formulation ($\mathbf{u} - \boldsymbol{\omega}$). There is now a set of either 2 or 3 Poisson equations to be solved, depending on the dimension of the system,

$$\nabla^2 \mathbf{u} = \nabla \times \boldsymbol{\omega}$$

Clearly this does not require any further differentiation to arrive at particle velocities. Calculation of boundary conditions is also much simpler in most cases, as Neumann conditions will reduce to simpler Dirichlet BCs. While this formulation was chosen for all calculations in the current study, it is important to note that the continuity equation is only implicitly satisfied, whereas it is part of the definition of the stream-function.



- Solve Poisson equation $\rightarrow \mathbf{u}_m$
- Calculate particle velocities $\mathbf{u}_p$ using previous area-weighting

Figure 2.6: Steps 2 & 3 of PIC algorithm – Solve, Convect

To finish the calculation loop, the same interpolation procedure that was used for the circulation spreading is now used to calculate the velocity at each particle location as a function of the velocities just calculated at the neighbouring mesh nodes, (See Fig. 2.6).

2.2.1 Three dimensions

The first application of PIC-type techniques for calculation of three-dimensional flows was by Couët *et al*, [21]. Using filaments rather than particles, discrete marker points are used to represent the vortex filaments, and these points are convected according to the induced velocity field. This field is calculated in an identical manner to Christiansen's original PIC technique. Poisson solver calculations are performed in Fourier space to aid computational efficiency, a modification described in detail in § 4.1.1. The stretching term is dealt with by enforcing that point vorticity vectors remain tangent to the filaments, and by ensuring the vorticity magnitudes are altered according to the local stretch at the point locations.

The first attempt at actual three-dimensional particle-in-cell was by Brecht

& Ferrante for studies of inviscid buoyant bubbles [22][23]. Table.2.1 summarises major publications in the field to date.

The majority of these publications rely on a combination of qualitative comparison to experimental data and conservation of invariants (§.A.3) to decide the relative success or failure of their techniques. It is only more recent work, [24] that has performed simulations which can be compared in a more quantitative way with existing experimental and numerical data.

Author(s)	Year	Re	Simulations	Vel. Calcs.
Rehbach	1978	inviscid	Shed vortex sheet	Direct
Couët *et al.*	1981	inviscid	CHECK	PIC
Brecht & Ferrante	1989	invsicid	Buoyant bubbles	PIC
Knio & Ghoniem	1990	inviscid	Perturbed vortex ring	CHECK
Winckelmans & Leonard	1993	400, inviscid	Colliding vortex rings	Direct
Ould-Salihi *et al.*	2000	1400	Impinging vortex ring	PIC
Liu & Doorly	2000	1000, inviscid	Colliding vortex rings	PIC
Walther & Koumoutsakos	2001	180, 230	Falling drop	PIC
Ploumhans & Winckelmans	2002	300, 500, 1000	Impulsively started sphere	Tree Code
Huberson & Voutsinas	2002	inviscid	Shed vortex sheet	PIC
Cottet & Poncet [24]	2004	300, 400	Impulsively started cylinder / Impinging vortex ring	PIC

Table 2.1: Chronological summary of three-dimensional vortex simulations

Chapter 3

Fundamental Concepts

3.1 Lagrange vs. Euler

There are two approaches to representing fluid flows, both of which are incorporated in this work. The first, *Eulerian*, approach, uses the concept of a continuum field, fixed in space, to represent the fluid's velocity, pressure, etc. This method gives information about the flow in terms of changes at fixed points in space as the fluid passes over these points. Traditional CFD techniques; finite volume, finite difference and finite element use this method almost exclusively, and it forms

the basis for major commercial codes.

The second, *Lagrangian*, method follows the positions of fluid particles as they move in the fluid, flow information is now stored on the particles. The values of the various fluid properties at these particle locations are determined as they convect in the flow, and in this way the behaviour of the flow is captured.

Both techniques have inherent advantages and disadvantages. To solve for a field variable in a certain domain, an Eulerian approach superimposes a grid over the entire domain. The boundaries of this grid usually have to extend outwards quite substantially, in order to avoid spurious interactions between the flow and the grid boundaries. Lagrangian techniques on the other hand, need only ever place particles where they are required, so far fewer calculations are required.

Fortunately, several types of fast solvers have been developed for use on fixed grids, so this speed-up usually far outweighs the advantage of smaller solution set requirements in the Lagrangian approach. However, in situations where the distribution of the field property in question is very compact, (i.e. wing-tip vortices, free jets) this may not always be the case, as the majority of calculations in the Eulerian approach will be redundant. A quantitative comparison of this issue was illustrated in Fig. 2.3.

Most CFD techniques often require the ability to match computational point density to gradients of one or more of the flow properties being examined, both to ensure solution convergence and accuracy. An example of this requirement is the resolution of a turbulent boundary layer, where, due to the extremely high velocity gradients normal to the wall, very high mesh density is required for Eulerian velocity field calculations. This mesh refinement becomes extremely laborious, especially in three-dimensions, and also requires a reasonably accurate knowledge of the expected flow-field *a priori*.

With Lagrangian techniques, free particles are the computation points, and it is a trivial operation to alter their density to match the solution requirements. A further advantage is that the particles will largely follow the gradients of the

flow in which they are convecting, giving a crude form of self-refinement.

3.2 Vorticity

The motion of a fluid is described by the velocity field $\mathbf{u}(\mathbf{x}, t)$. The curl of the velocity field is called the vorticity, $\boldsymbol{\omega}(\mathbf{x}, t)$. In 3D cartesian coordinates,

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} = \begin{bmatrix} \dfrac{\partial w}{\partial y} - \dfrac{\partial v}{\partial z} \\ \dfrac{\partial u}{\partial z} - \dfrac{\partial w}{\partial x} \\ \dfrac{\partial v}{\partial x} - \dfrac{\partial u}{\partial y} \end{bmatrix} \tag{3.1}$$

It follows from the definition that the vorticity is solenoidal, i.e. there cannot be divergence present in the vorticity field,

$$\nabla \cdot \boldsymbol{\omega} = 0 \tag{3.2}$$

Physically, vorticity is a measure of the rotational characteristics of a flow. For a infinitesimal fluid element in that flow, its vorticity is twice its average angular velocity. The use of this property as the basis for a computational technique is obvious, and it is this link that is the starting point of all vortex methods.

A property related to the vorticity field in a similar way that kinetic energy is related to the velocity field is the *enstrophy*, $\mathcal{E}$ which is simply the square of the vorticity, $\boldsymbol{\omega} \cdot \boldsymbol{\omega}$. This property is used as a measure of the total vorticity present in the flow, and the change in total enstrophy with time is an important diagnostic in analysing the amount of vortex stretching acting on the vorticity. This behaviour is described in more detail in A. A.3.

Two other properties of the flow which are important for use in analysis of the conservation properties of the flow. These are *linear* and *angular* impulse. For unbounded three-dimensional flows at rest at infinity, they should both be

invariant, even with viscous effects present in the flow. The linear impulse is defined as

$$\boldsymbol{I} = \frac{1}{2}\int \mathbf{x} \times \boldsymbol{\omega} \mathrm{d}x \tag{3.3}$$

It is shown in Saffman [25] that this plays the part of momentum in unbounded flows, because

$$\frac{\mathrm{d}\boldsymbol{I}}{\mathrm{d}t} = \int F \, \mathrm{d}x \tag{3.4}$$

where F is the sum of the non-conservative external body forces. Clearly, if the right hand side is zero then there can be no change in the value of $\boldsymbol{I}$ even when the flow is unsteady. Also, for jet flows, the value of $\boldsymbol{I}$ should increase linearly as a function only of the jet velocity and nozzle area

Knowledge of the vorticity means that the velocity field is known implicitly, and vice versa. The vorticity field can be recovered from a velocity field simply from its definition Eq. 3.1. Calculation of the velocity field from a given vorticity distribution requires either the solution of the *Biot-Savart* law, or a Poisson equation in either velocity or stream-function form.

The exact form of the Biot-Savart law differs depending on the dimension of the problem,

$$\mathbf{u} = \int \mathbf{K}(\mathbf{x}-\mathbf{y}) \times \boldsymbol{\omega}(\mathbf{y}) d\mathbf{y} \tag{3.5}$$

$$\mathbf{K}(\mathbf{z}) = \begin{cases} -\frac{1}{2\pi}\mathbf{z}/|\mathbf{z}|^2 & \text{in two dimensions} \\ \frac{1}{4\pi}\mathbf{z}/|\mathbf{z}|^3 & \text{in three dimensions} \end{cases} \tag{3.6}$$

In purely Lagrangian solvers, the Biot-Savart law is discretised to give a summation equation for the velocities induced by a set of finite vortex particles. In particle-in-cell techniques, a Poisson equation is solved on the fixed grid in order to find the particle velocities.

3.2.1 Points, Blobs, Sheets, Filaments *etc.*

It is relevant at this stage to give a brief summary of the terminology associated with various vortex structures. In two dimensions there are three main types,

Points Point vortices are singular distributions of vorticity, where the vorticity is represented by a Dirac Delta function

Blobs By convolving a point vortex with a smoothing function, the vorticity distribution is spread outwards, usually in the form of a Gaussian.

Sheets Vortex sheets are singular distributions along a line, as opposed to on a point.

In three dimensions the situation becomes more interesting.

Points Identical to singular particle in 2D, except vorticity now represented by a vector.

Blobs 2D smoothing functions can be extended to 3D, giving a natural equivalent.

Vortex line Just as a streamline is a curve which is always tangent to the velocity field, can define a *vortex line* as a curve which is always tangent to the local vorticity field.

Vortex tube The surface that is formed by all the vortex lines passing through some closed curve in space. A schematic of a vortex tube made up of several vortex lines is shown in Fig. 3.1.

Vortex ring If a vortex tube forms a closed loop, it is termed a *vortex ring*.

Vortex filament If the cross-section of a vortex tube is made infinitesimally small, it becomes a *vortex filament*, an important construct in three-dimensional vortex methods.

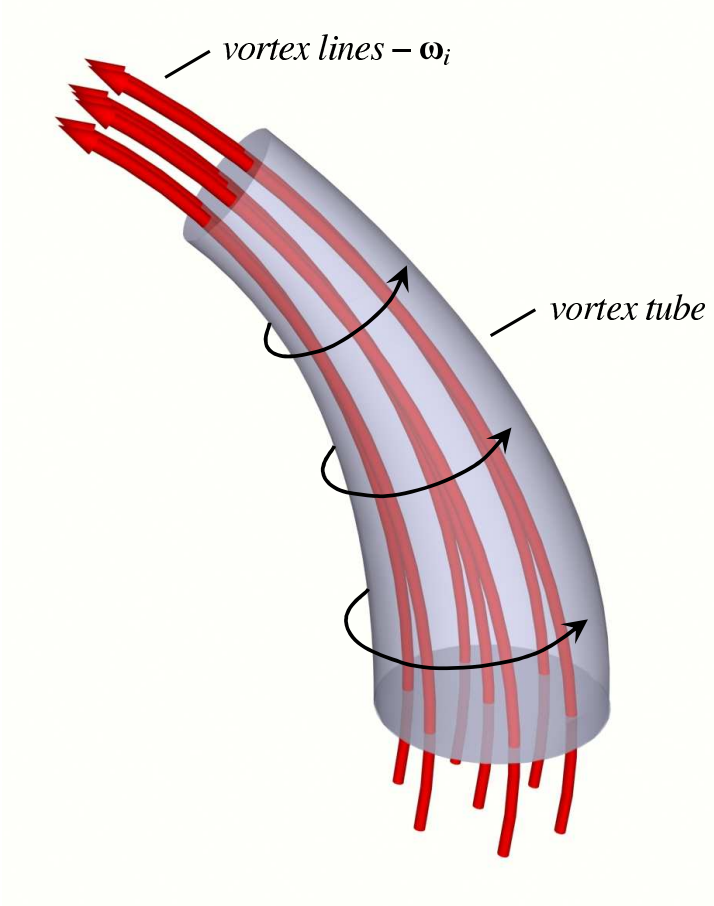


Figure 3.1: Section of vortex tube showing vortex lines

3.2.2 Circulation, Γ

A scalar property of importance in the description of vortex flows is the *circulation*, Γ. Whereas the vorticity describes rotation at a point, circulation describes the total rotation over a defined region. It can be defined in two ways. The first is as an area integral over a given vorticity distribution

$$\Gamma = \int_A \boldsymbol{\omega} \cdot \mathrm{d}A \tag{3.7}$$

From application of Stokes' theorem, circulation can also be defined as the line integral of velocity around a closed curve enclosing the region of vorticity

$$\Gamma = \oint_C \mathbf{u} \cdot \mathrm{d}s \tag{3.8}$$

Sign convection for vorticity and circulation obeys the right-hand rule, i.e. anticlockwise circulation is positive.

A section of vortex tube, shown in Fig. 3.2, is used to demonstrate the evaluation of the circulation of a finite region of vorticity. The circulation cannot change along the length of the tube, from Helmholtz's first law, (*see §3.4.1*).

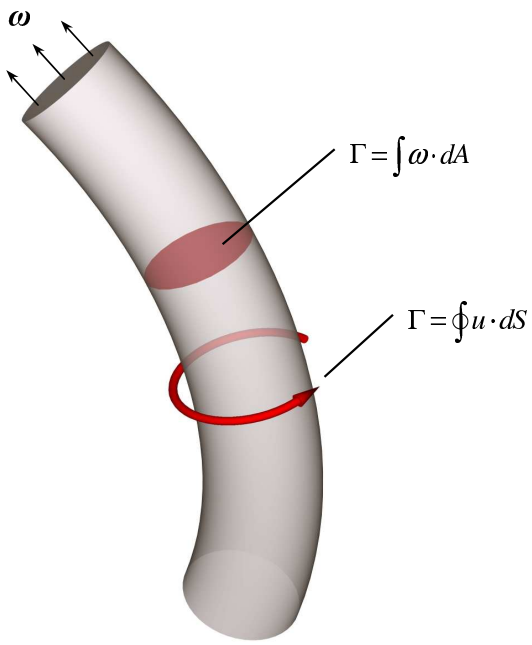


FIGURE 3.2: Section of vortex tube showing circulation Γ

Another property, related to the circulation, is the *vortex strength*, $\boldsymbol{\alpha}$. It has units of vorticity × volume.

$$\alpha = \int_V \boldsymbol{\omega} \cdot \mathrm{d}V \tag{3.9}$$

If a vorticity-carrying fluid element is thought of as a small section of a vortex tube, as shown in Fig. 3.3, it can also be defined as circulation × length–where the length is of the section of tube the element represents.

This property only has relevance in the context of three-dimensional vortex particles.

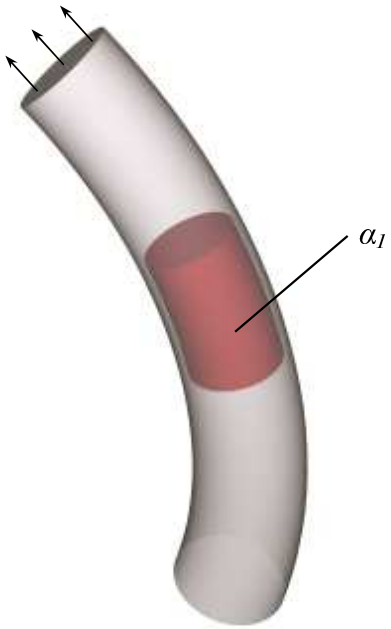


Figure 3.3: Section of vortex tube showing vortex strength $\boldsymbol{\alpha}$

3.3 Velocity gradients

Spatial velocity gradients are important for two reasons. Firstly, they govern the behaviour of the non-linear stretching term in the momentum equation. Secondly, they give a physical description of the deformation of the particles behaving as fluid elements.

The velocity gradient tensor, present in the stretching term in Eq. 3.21, can be written in full as

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} \tag{3.10}$$

The effect of each component on an initially cubic fluid element is illustrated in Fig. 3.4.

Any second-order tensor can be decomposed into two separate tensors, one symmetric and the other skew-symmetric, using the identity

$$\nabla \mathbf{u} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \tag{3.11}$$

These two new tensors, widely used in fluid dynamics, are identified as the *rate-of-strain* and *rotation* tensors.

$$\begin{aligned} \mathbb{S} &= \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \qquad \text{Rate of strain tensor} \\ \mathbb{R} &= \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T) \qquad \text{Rotation tensor} \end{aligned} \tag{3.12}$$

(3.13)

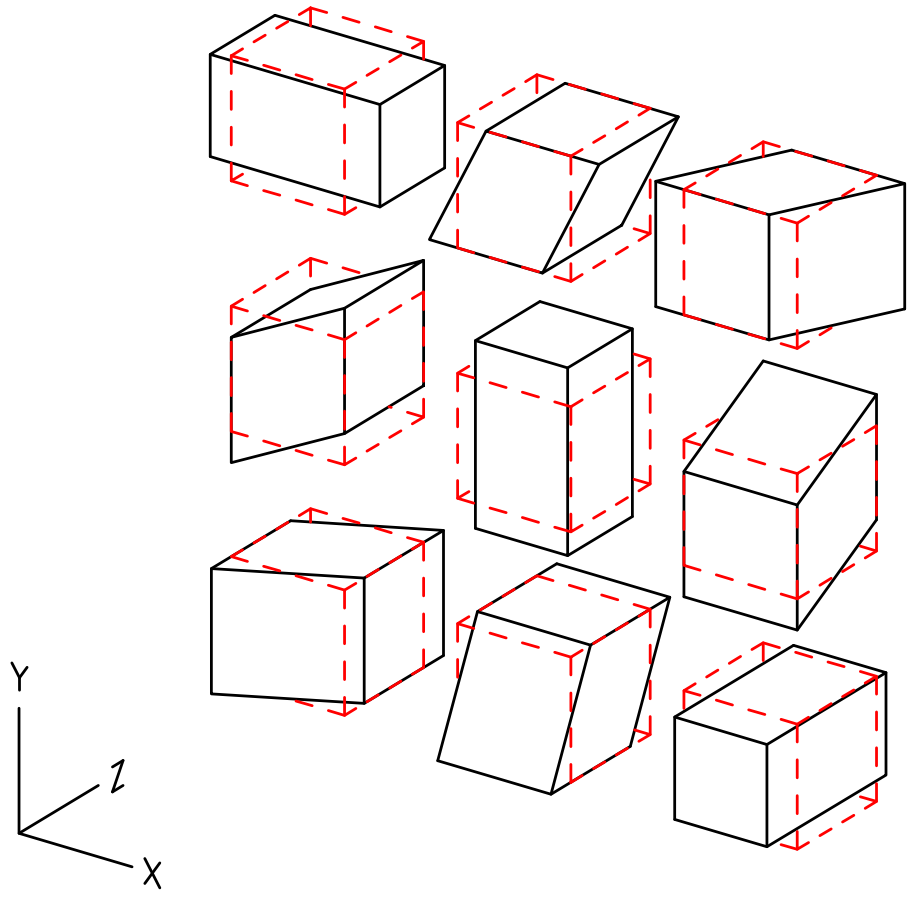


Figure 3.4: Stretching and tilting of fluid element by $\nabla \mathbf{u}$

Using the definition of vorticity, $\mathbb{R}$ can be rewritten as

$$\mathbb{R} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3.14}$$

In calculation of the stretching term in Eq. 3.21, the product of this tensor with the vorticity vector is identically zero, therefore only the rate-of-strain tensor contributes to changes in the vorticity field through stretching.

When investigating the contribution from $\mathbb{S}$, it is useful to further subdivide it into two further tensors, one containing the diagonal components, and the other containing just the off-diagonal components.

$$\mathbb{S} = \begin{bmatrix} \frac{\partial u}{\partial x} & 0 & 0 \\ 0 & \frac{\partial v}{\partial y} & 0 \\ 0 & 0 & \frac{\partial w}{\partial z} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 0 & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} & 0 \end{bmatrix} \tag{3.15}$$

As the flow is incompressible, the trace of the first tensor is zero. The three terms in this tensor correspond to stretching of the fluid elements in each of the three directions, as shown on the diagonal in Fig. 3.4. A positive value of one of these terms will cause an amplification of the existing vorticity component in the same direction when calculating the stretching term.

The second tensor contains information on how the fluid elements are being sheared by the local velocity gradients. There are only three independent terms, each modifying two of the vorticity components in calculation of the stretching term.

3.4 Vorticity dynamics

The incompressible Navier-Stokes equations are the starting point for the derivation of the equations governing the vorticity dynamics.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{3.16}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{3.17}$$

Where $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $\boldsymbol{\omega}(\mathbf{x}, t)$ is the vorticity field, p is the pressure, ρ the density and ν is the kinematic viscosity.

The problem with using this set of equations as the basis for a computational technique is that it is necessary to solve for both the pressure and velocity components at every time step. As this is a nonlinear system, this requires iteration at every time-step to achieve the solution, with strict convergence criteria.

The vorticity transport equation is derived by rewriting Eq. 3.16 as,

$$\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\omega} \times \mathbf{u} = -\nabla(\frac{p}{\rho} + \frac{\mathbf{u} \cdot \mathbf{u}}{2}) + \nu \nabla^2 \mathbf{u} \tag{3.18}$$

And then taking the curl of the resulting equation.

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \times (\boldsymbol{\omega} \times \mathbf{u}) = \nu \nabla^2 \boldsymbol{\omega} \tag{3.19}$$

The nonlinear term can be replaced using the identity

$$\nabla \times (\boldsymbol{\omega} \times \mathbf{u}) = -(\mathbf{u} \cdot \nabla)\boldsymbol{\omega} + (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} \tag{3.20}$$

Note that this identity only holds if $\nabla \cdot \mathbf{u} = 0$ and $\nabla \cdot \boldsymbol{\omega} = 0$. While this does implicitly incorporate the continuity equation into the vorticity transport equation, it will be shown in § 4.5 that the vorticity field represented by a discrete set of particles cannot be guaranteed to remain divergence free for all times.

The vorticity transport equation can now be written in the familiar form,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} \tag{3.21}$$

Rate of change of $\boldsymbol{\omega}$	Convection	Stretching	Viscous diffusion

The discretisation of the three transport terms in Eq. 3.21, convection, stretching and diffusion is described in § 4.1–§ 4.3.

3.4.1 Helmholtz's and Kelvin's Laws

Work published by Hermann von Helmholtz (1821-1894) in 1858, and by William Thomson, later Lord Kelvin (1824-1907) in 1869 initiated the mathematical study of vortex motion. They produced a set of laws that govern the behaviour of vorticity-carrying regions in a flow.

It is worth referring back to the diagram of the vortex tube in Fig. 3.1 in the previous section, as these laws are most readily explained in relation to such a structure.

Helmholtz's first law states that, for inviscid flows, the *substantive* derivative of the circulation must always be equal to zero.

$$\frac{\mathrm{D}\Gamma}{\mathrm{D}t} = 0 \tag{3.22}$$

The substantive, or material, derivative is used to describe time rates of change in a Lagrangian frame, as it accounts for both standard variations in time, and also variations due to convection ($\mathrm{D}/\mathrm{D}t = \partial/\partial t + \mathbf{u} \cdot \nabla$).

A consequence of this law mean that fluid elements that at any instant lie inside a vortex tube will remain within the tube, independent of velocity gradients acting on the tube. A useful application of this law is that vortex filaments can be tracked through a velocity field by simply monitoring the material line corresponding to the initial filament centreline.

The inverse of this law means that any particle initially free of vorticity must remain irrotational for all consequent times.

Helmholtz's second law states that the circulation of a vortex tube must be the same at all cross sections. Therefore, if the tube narrows due to local velocity gradients, the vorticity must increase.

This law also restricts the possible configurations of vortex lines and tubes. These structures must be stretch to infinity, end at a boundary, or form a closed loop. These three allowable configurations are shown in Fig. 3.5.

Kelvin's law extends Helmholtz's first law to include the effect of viscosity. For the circulation of a vortex tube in a viscous flow,

$$\frac{\mathrm{D}\Gamma}{\mathrm{D}t} = \nu \int_L (\nabla^2 \mathbf{u}) \cdot d\mathbf{r} \tag{3.23}$$

Clearly, for inviscid flow, the RHS of Eq. 3.23 is zero, thus reducing to Helmholtz's first law.

An interesting aside in Helmholtz's publication was one of the first descriptions of leapfrogging vortex rings. He described his conclusions regarding two circular vortex rings with a common axis of symmetry as follows,

> *If they both have the same direction of rotation they will proceed in the same sense, and the ring in front will enlarge itself and move slower, while the second one will shrink and move faster, if the velocities of*

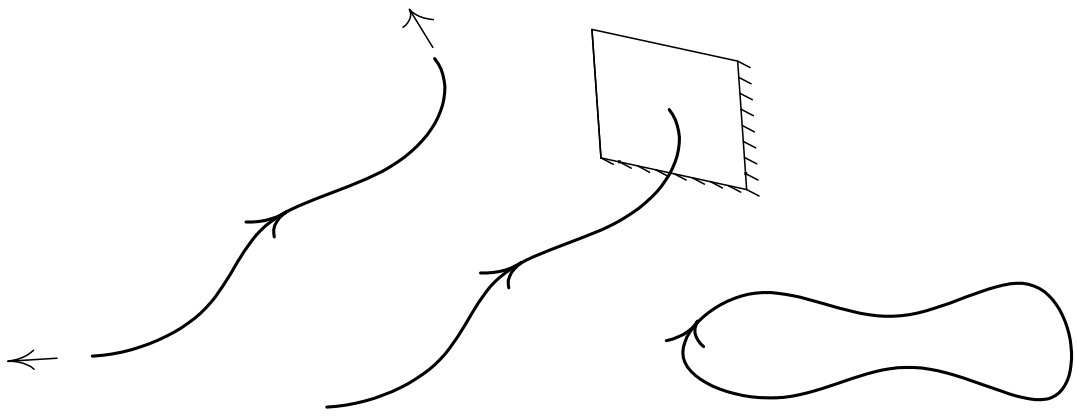


Figure 3.5: Allowable vortex line configurations

> *translation are not too different, the second will finally reach the first and pass through it. Then the same game will be repeated with the other ring, so the rings will pass alternately one through the other.*

An attempt at modelling this phenomenon is described in § 6.3. It is quite difficult to model the 'game' for long enough to capture the pass of the second ring back through the first ring, as this requires to maintain a near-inviscid flow regime for a relatively long simulation time. However, as will be demonstrated later, some diffusion is necessary to maintain solution stability.

It is not just numerical models that have problems with this flow however, as to date there has only been one successful experimental study of this 'double-jump' published [1].

Chapter 4

Numerical Modelling

The technique developed in this work is fundamentally an extension of the Particle-In-Cell method first applied to the modelling of fluid flows by Christiansen [12]. A hybrid Eulerian-Lagrangian approach is used to solve the vor-

ticity transport equation Eq. 3.21. In each time step, the vortex particles are convected with their local velocities, and their strengths are modified to account for stretching and diffusion. All field calculations are performed on the fixed Eulerian grid, and this information is transferred to the particles in order to advance the solution in time.

§ 4.1 deals with treatment of the convection term in the vorticity transport equation, and the tools required to model this term. § 4.2 deals with the stretching term, § 4.3 with viscous diffusion. § 4.4 describes how the solution is advanced in time. § 4.5 discusses the problem of vorticity divergence related to three dimensional vortex particle methods, and finally § 4.6 gives a brief discussion of errors arising in calculation of each of the terms.

4.1 Convection

The use of vortex methods to represent the convection term in the vorticity transport equation is made possible by Kelvin's theorem. This states that the circulation contained in a Lagrangian fluid element is conserved, provided that the flow is inviscid and there are no external forces. Therefore, the temporal development of a set of marker particles carrying concentrated vortex strength, equivalent to an initial vorticity field, can be used to represent the actual fluid dynamics. The convection term $(\mathbf{u}\cdot\nabla)\boldsymbol{\omega}$ in Eq. 3.21 is thus not treated explicitly, as it is absorbed into the Lagrangian material derivative on the particles.

4.1.1 Poisson Solver

The velocity field is calculated from the vorticity field using a system of Poisson equations

$$\nabla^2\mathbf{u} = -\nabla\times\boldsymbol{\omega} \tag{4.1}$$

This equation set is derived by appling the curl operator to the vorticity

definition Eq. 3.1, and using the vector identity

$$\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u} \tag{4.2}$$

Use of the incompressibility constraint on the velocity field, ($\nabla \cdot \mathbf{u} = 0$) is also required.

There are several approaches available for solution of this equation set. The simplest of these is finite difference on a regular structured mesh, where a standard set of stencils are used to calculate the required gradients in equationeq:Poisson, and a set of linear equations is solved. Fast Fourier transforms can be used to reduce the computational cost of this approach from $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2 \log M)$, where M is the number of mesh nodes.

More advanced techniques use coordinate mapping on the initially regular grid to provide finer mesh density in areas of high spatial gradients, or an entirely unstructured mesh is used. Clearly these approaches are much more computationally demanding that the optimised FFT solver using the regular mesh, so it was decided to use the simplest solver in the current study. Another factor in this decision was that development of a fast Poisson solver was outside the scope of this work, and there are several open-source FFT solver codes available which can solve the specified equation set. `fishpack` [26] is used in this instance. Using the RHS as input, the following steps are performed.

1	Discrete Fourier Transform in x direction
2	Discrete Fourier Transform in y direction
3	Tridiagonal Solve in z direction
4	Inverse Transform in y direction
5	Inverse Transform in x direction

Table 4.1: Steps in FFT of Poisson solver

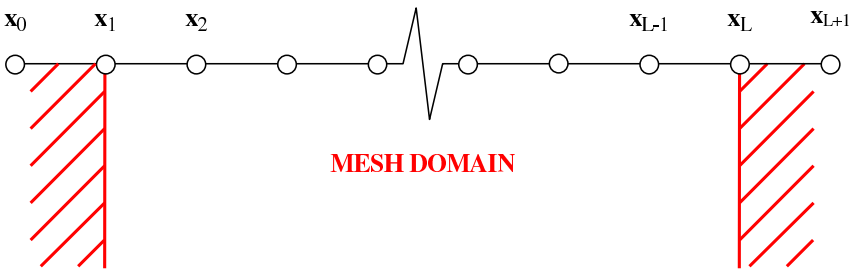
The following system of linear equations is solved by Fishpack.

$$\begin{aligned}
&\mathbb{C}_1 \cdot (\mathbf{x}_{i-1,j,k} - 2\mathbf{x}_{i,j,k} + \mathbf{x}_{i+1,j,k}) + \\
&\mathbb{C}_2 \cdot (\mathbf{x}_{i,j-1,k} - 2\mathbf{x}_{i,j,k} + \mathbf{x}_{i,j+1,k}) + \\
\mathbf{A}(k) \cdot \mathbf{x}_{i,j,k-1} + \mathbf{B}(k) \cdot \mathbf{x}_{i,j,k} + \mathbf{C}(k) \cdot \mathbf{x}_{i,j,k+1} &= \mathbf{F}(i,j,k) \\
&\text{for } i = 1 \ldots L,\ j = 1 \ldots M,\ k = 1 \ldots N \quad (4.3)
\end{aligned}$$

where L, M and N are the number of grid nodes in the x, y and z directions respectively.

This represents a standard second-order centered finite difference of the Laplacian operator. The right hand side (*i.e.* the source term) of Eq. 4.3 is generated using the 4th-order finite difference stencils derived in A.2.

A limitation of the FFT approach is that some form of periodicity must be assumed in the x and y directions. The full set of options for the x- and y-direction periodicity boundary conditions are listed in table 4.2.



Periodic Options		
0	$\mathbf{x}_0 = \mathbf{x}_L$	$\mathbf{x}_{L+1} = \mathbf{x}_1$
1	$\mathbf{x}_0 = 0$	$\mathbf{x}_{L+1} = 0$
2	$\mathbf{x}_0 = 0$	$\mathbf{x}_{L+1} = \mathbf{x}_{L-1}$
3	$\mathbf{x}_0 = \mathbf{x}_2$	$\mathbf{x}_{L+1} = \mathbf{x}_{L-1}$
4	$\mathbf{x}_0 = \mathbf{x}_2$	$\mathbf{x}_{L+1} = 0$

Table 4.2: Options for periodic boundary condition (LPEROD) in Poisson solver

Analysis of the effect of each of these periodicity constraints of the calculated velocity field showed that case 0 had the least influence on the flow field. So for the x direction,

$$\begin{aligned} \mathbf{x}(0, j, k) &= \mathbf{x}(L, j, k) \\ \mathbf{x}(L+1, j, k) &= \mathbf{x}(1, j, k) \end{aligned} \tag{4.4}$$

and similarly for the y direction. The exact form of this periodicity is shown in Fig. 4.1.1.

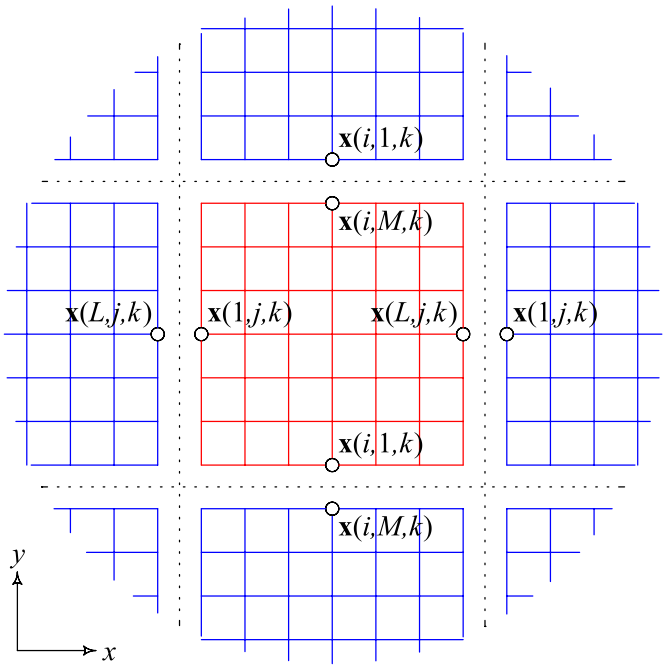


Figure 4.1: Solution periodicity in x and y directions

There is no such periodicity restriction in the z direction, so any boundary conditions can be applied. This allows convection of vorticity into and out of the domain–an obvious requirement in the simulation of free jets.

In Eq. 4.1, Neumann boundary conditions are applied to velocities at the inflow. The outflow requires special treatment as vorticity must be allowed to

convect out of the domain without any spurious feedback on the upstream flow. The following expression is used to 'decay' the vorticity at the outflow

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \mathbb{C}\frac{\partial \boldsymbol{\omega}}{\partial z} \tag{4.5}$$

This is the *convective* outflow boundary condition used widely in large eddy simulations. The value used for $\mathbb{C}$ is the local convection velocity, although it has been found that the effect on upstream vorticity is largely insensitive to choice of this value. Orlandi [27] uses a constant value of 0.6 of the normalised characteristic velocity, and solution insensitivity was again reported.

While all calculations in this solver are performed in single precision (*i.e.* 32 bit), it was established in analysis of error magnitudes that errors induced from this truncation were negligible in comparison to other errors (*see* § 4.6).

4.2 Stretching term

The stretching term in the vorticity transport equation causes local intensification and reorientation of the vorticity vector, governed by shearing forces acting on the vorticity field. This effect is not present in two-dimensional flows and is challenging to accurately model.

Particle strengths are modified according to the local discretisation of the stretching term $(\boldsymbol{\omega}\cdot\nabla)\mathbf{u}$. The velocity gradient tensor $\nabla\mathbf{u}((x)_p$ is evaluated on the grid using finite differences. This tensor is then interpolated on to the particle locations using the standard B-spline interpolation (see § A.1). Particle vorticities can now be updated, as in a grid-free vortex method, by solving

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = [\nabla\mathbf{u}((x)_p][\boldsymbol{\omega}_p] \tag{4.6}$$

where $\boldsymbol{\omega}_p$ is the vorticity at a particle location. High-order finite difference stencils are used throughout, in an effort to avoid numerical error and numerical dissipative effects.

Another possibility which was investigated, derived by Cottet in [28], is based on the conservative form $\mathrm{div}(\boldsymbol{\omega} : \mathbf{u})$ of the stretching term. The idea is to multiply grid values of vorticity and velocity. The tensor $\omega_i u_j$ is then differentiated on the grid, and its divergence is finally interpolated on the particles to update their circulations. This last option has the advantage of being conservative at the grid level, even when the vorticity field is not exactly divergence-free. However, the classical formulation where the velocity gradient tensor is spread directly on to the particles was considered a simpler approach and is used throughout in calculation of the stretching term.

A major advantage of the $\mathbf{u} - \boldsymbol{\omega}$ formulation of the Poisson equation over the $\Psi - \boldsymbol{\omega}$ form becomes apparent here. If the stream-function were solved on the grid, it would have to be differentiated once to calculate velocities, and a second time to calculate the velocity gradient tensor. However, if the velocity field is calculated directly by the Poisson solver using Eq. 4.1, only one level of differentiation is required to extract the required gradients, leading to a higher level of accuracy.

Any numerical error introduced in calculation of this term will introduce divergence into the vorticity field. If singularities in the vorticity field are allowed to develop unchecked, they will eventually cause the solution to become unstable. This is the greatest weakness of three-dimensional vortex particle methods, and requires that much greater care be taken in maintaining solution accuracy than would be necessary in a corresponding two-dimensional technique. A more detailed discussion of this error and means of counteracting it are given in § 4.5.

The most relevant diagnostic for evaluating the accuracy of this technique is the angular momentum in the system. So if a vortex structure is stretched or reorientated, while the vorticity field undergoes large changes, the angular momentum remains constant and non-zero. For simple flows an estimate of this value can be obtained *a priori*. Further discussion of the behaviour of this diagnostic is given in chapter 6.

4.2.1 Analytical solution

An analytical expression for stretching term is derived in [29] for three-dimensional particles with core models,

$$\frac{d}{dt}\boldsymbol{\alpha}^p = -\sum_q \frac{1}{\sigma^3}\left[\frac{g(\rho)}{\rho^3}\boldsymbol{\alpha}^p \times \boldsymbol{\alpha}^q + \frac{1}{\sigma^2\rho}\frac{d}{d\rho}\left(\frac{g(\rho)}{\rho^3}\right)(\boldsymbol{\alpha}^p \cdot (\mathbf{x}^p - \mathbf{x}^q))((\mathbf{x}^p - \mathbf{x}^q) \times \boldsymbol{\alpha}^q)\right] \tag{4.7}$$

where $\rho = |\mathbf{x}^p - \mathbf{x}^q|/\sigma$ is the distance between particles normalised by core width.

The case of singular vortex particles is derived by setting $g(\rho) = 1$ and $\frac{1}{\rho}\frac{d}{d\rho}(\frac{g(\rho)}{\rho^3}) = \frac{-3}{\rho^5}$. The term $q = p$ must also be excluded from the summation. Eq. 4.7 now becomes,

$$\frac{d}{dt}\boldsymbol{\alpha}^p = \sum_q^{q\neq p} \frac{3}{|\mathbf{x}^p - \mathbf{x}^q|^5}\big(\boldsymbol{\alpha}^p \cdot (\mathbf{x}^p - \mathbf{x}^q)\big)\big((\mathbf{x}^p - \mathbf{x}^q) \times \boldsymbol{\alpha}^q\big) - \frac{1}{|\mathbf{x}^p - \mathbf{x}^q|^3}\boldsymbol{\alpha}^p \times \boldsymbol{\alpha}^q \tag{4.8}$$

This expression is used in calculation of the error introduced in the mesh-based calculation of the stretching term.

4.3 Diffusion

As outlined in the literature review, it is very important to accurately model viscous effects in real-world flows. While several approaches were introduced, it was found that mesh-based evaluation of the viscous diffusion performed adequately for the flows of interest in this work, and was also much less computationally expensive than the particle strength exchange technique.

Calculation of the viscous diffusion requires evaluation of the Laplacian in Eq. 3.21. Peaks in the vorticity field will have a high value of the Laplacian, and so viscosity will have the highest effect on such regions.

The *Particle Strength Exchange* technique replaces the Laplacian with an integral operator, not requiring regular particle spacing. The problem now becomes

$\mathcal{O}(N^2)$, so requires some form of multipole expansion, in order to achieve some level of computational efficiency.

Fortunately, where a fixed grid is used, $\nabla^2\boldsymbol{\omega}$ can be solved directly on the grid using finite differences. The result can then be spread directly back onto the particles using the standard interpolation technique, and the particle strengths modified accordingly.

Clearly, the spatial resolution of the technique is limited by the mesh resolution, and the scheme does not implicitly conserve circulation. However, simulations using this technique, coupled with frequent regridding, maintain circulation sufficiently well for this shortcoming not to be noticeable.

For high Reynolds number flows, it is only necessary to calculate the viscous diffusion term at integer multiples of the time-step. This is because, in term of overall effect on circulation magnitude, it has a minor effect in comparison to the stretching term.

4.4 Temporal Discretisation

In order to advance the solution in time, the convection and stretching term are coupled over several steps of a Runge-Kutta time-stepping algorithm.

Third-order Low-storage Runge-Kutta is used [30]. It is used to calculate both the change in particle location due to the induced velocity field, and to calculate the change in particle strength vector due to the stretching term. Though not as accurate as the classical fourth order Runge-Kutta technique, it has the advantage that only two memory registers are required during the solution, as opposed to five for the classical approach. This is made possible by overwriting intermediate values at each solution substep.

$$
\begin{aligned}
q_1 &= \Delta t \cdot \mathbf{u}(\mathbf{x}_0) - 6e_3 \\
\mathbf{x}_1 &= \mathbf{x}_0 + \frac{1}{3}q_1 \\
q_2 &= \Delta t \cdot \mathbf{u}(\mathbf{x}_1) - \frac{10}{3}e_1 - \frac{5}{9}q_1 \\
\mathbf{x}_2 &= \mathbf{x}_1 + \frac{15}{16}q_2 \\
q_3 &= \Delta t \cdot \mathbf{u}(\mathbf{x}_2) - \frac{15}{8}e_2 - \frac{153}{128}q_2 \\
\mathbf{x}_3 &= \mathbf{x}_2 + \frac{8}{15}q_3
\end{aligned} \tag{4.9}
$$

Where e_i are round-off errors, defined as,

$$
\begin{aligned}
e_1 &= (x_1 - x_0) - \frac{1}{3}q_1 \\
e_2 &= (x_2 - x_1) - \frac{15}{16}q_2 \\
e_3 &= (x_3 - x_2) - \frac{8}{15}eq_3
\end{aligned} \tag{4.10}
$$

The two memory registers are used to store $\mathbf{x}$ and q, though the algorithm can be modified to store $\mathbf{x}$ and $\mathbf{u}$, an approach more suited to memory requirements of other subroutines.

4.5 Vorticity Divergence

The problem with three-dimensional particle methods is that they are not inherently stable. This means that as a simulation is run, some form of error accumulates, usually leading to unphysical velocities, which eventually become singular. The errors are apparent in divergence present in the particle vorticity field.

Where the stretching term introduces strong local vorticity intensification, this leads to rapid distortion of the particle field. Regular redistribution of the particle strengths is thus required to maintain a smooth solution field. This

regridding is achieved using the same interpolation function as used to trasnfer data to and from the mesh to the particles. A full description of this function is given in § A.1.

A meaningful physical interpretation of this error is illustrated in Fig. 4.2. This shows a vortex filament before and after deformation due to velocity gradients. Helmholtz's laws state that the circulation must remain constant about the circumference of the filament, and that vorticity vectors must remain tangent to the filament for all times. So where the filament is stretched, the vorticity vector increases in magnitude, and where it is distorted, the vorticity vector changes its orientation accordingly to remain tangent to the filament.

This reorientation and amplification is dealt with implicitly in vortex filament techniques. Using Helmholtz's second law, A filament can be represented by a spline or set of line segments, which are defined by a set of marker points. The vorticity carried by the marker points is altered depending on the reshaping of the lines on which they lie, an exact analogy of the physical situation.

In the Particle-in-Cell approach this filament is now represented by a set of *independent* particles carrying vortex strengths. The only effect these particles have on each other is through their induced velocities, there is no longer any geometric constraints. Because of this freedom, the particle strength vector is no longer constrained in any way by the underlying filament it is supposed to represent. Instead, its strength is altered according to the local velocity gradients, through the stretching term $[\nabla \mathbf{u}][\boldsymbol{\omega}]$. The hope is that this alteration will mimic the change in the filament.

So, what happens when there is an error in this stretching term? The effect of any errors is shown in the magnified section in Fig. 4.2. Blue vectors represent a solution by vortex filaments. Clearly, the magnitudes and orientation of these vectors change according to the deformation of the filament.

The red vector shows a possible particle solution. The magnitude is too large, violating the first law, and the vector is no longer tangent to the filament,

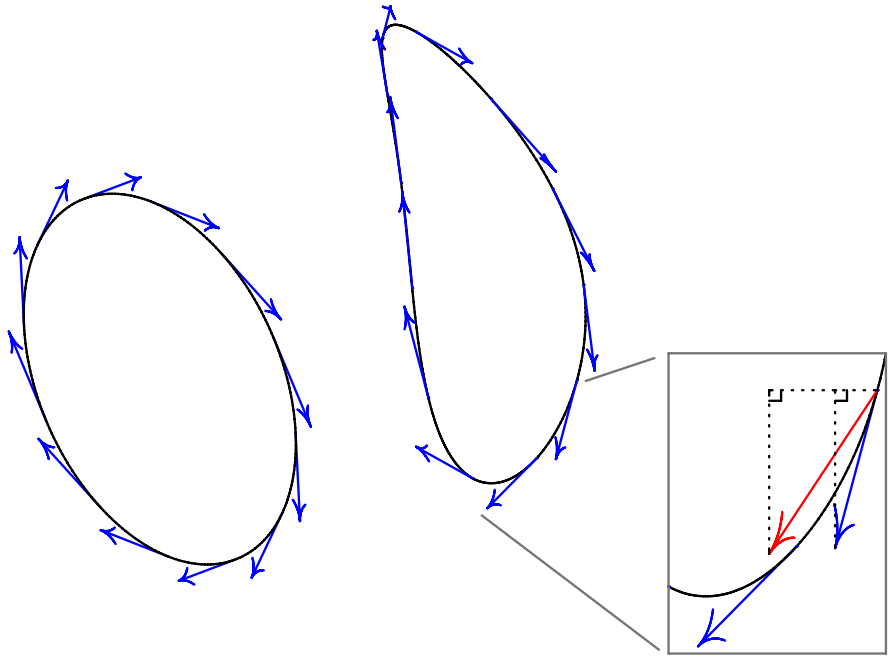


Figure 4.2: Introduction of errors through deformation of filaments

so violating the second law. It seems the largest source of error is in the off-diagonal terms in the velocity gradient tensor, as these are solely responsible for reorientation of the strength vector.

The vorticity field associated with a three-dimensional vortex particle technique cannot be guaranteed to remain divergence free for all times. Theoretically, this represents a major problem, as a basis which is not divergence free, $\boldsymbol{\omega}_p(\mathbf{x}, t)$ is being used to represent a vector field that is required to be divergence free for all times, according to Helmholtz's Laws. The particle method is only solving what it sees in accordance with its particular discretisation of the vorticity transport equation Eq. 3.21 and its representation of the initial vorticity field, which, depending on the coarseness of the solution, can differ greatly form the actual physics. If this difference is allowed to develop unchecked, the solution can become a very poor representation of the real vorticity field, and may eventually become unstable. This fact is in opposition with two-dimensional singular vortex particles (i.e. point vortices) for which the particle vorticity field is uncondi-

tionally divergence free, and is a serious obstacle to the validity of finitely many singular particles as a model for the solution of the three-dimensional vorticity equation.

In practice, fortunately, if this divergence is kept to a minimum, by careful solution of the stretching term and by regular reallocation of the vorticity onto a regular particle distribution, its effect on the solution can be reduced substantially. It is also critical to minimise divergence in the initial vorticity present in the flow, and any vorticity introduced at later times. So, in order to maintain regularity of solution for long runs, some technique is required to remove this divergence from the vorticity field, while maintaining the correct flow field information. Such techniques are discussed in the next section §4.5.1.

4.5.1 Regularisation

One of the first attempts at vorticity divergence removal was introduced in [31] for a purely Lagrangian technique. At every time-step the particle strength vectors are modified using the filtering

$$\boldsymbol{\alpha}_p^{new} = (1 - \mathbb{C})\boldsymbol{\alpha}_p^{old} + \mathbb{C}\frac{\nabla \times \mathbf{u}(\mathbf{x}_p)}{|\nabla \times \mathbf{u}(\mathbf{x}_p)|}|\boldsymbol{\alpha}_p^{old}| \tag{4.11}$$

where $\nabla \times \mathbf{u}(\mathbf{x}_p)$ is the true local vorticity field and where $\mathbb{C}$ is a relaxation factor. Physically, this scheme acts as a 'spring' that tries to maintain the particle strength vector aligned with the true vorticity vector. This is a very simple technique as no system of linear equations involving other locations needs to be solved.

However, this technique was developed, and proven satisfactory, in the context of purely Lagrangian techniques, and so does not account for extra errors involved in interpolating data back and forth from the grid.

A basic identity from vector calculus is that the curl of *any* vector field, is by definition divergence free, even if the initial field contains singularities. Clearly then, combining this property with the definition of the vorticity as the curl

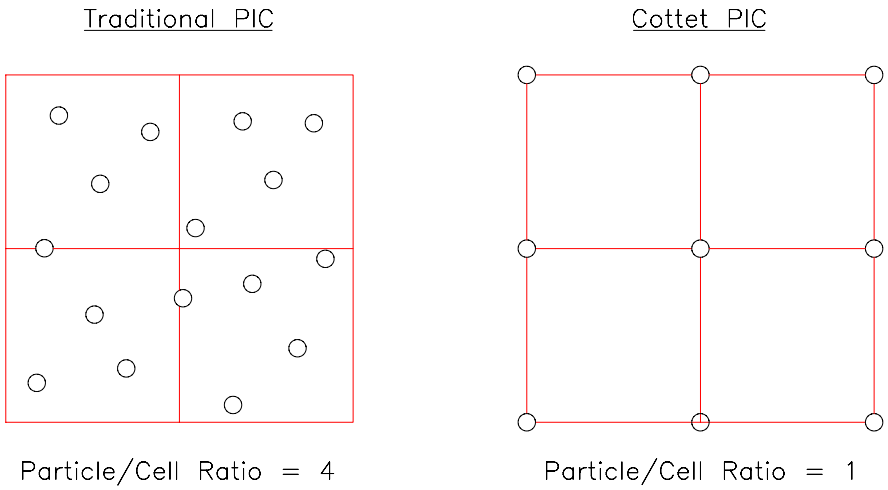


Figure 4.3: Comparison of traditional to Cottet PIC

of a velocity field gives an idea as to how to proceed with removal of vorticity divergence.

Unfortunately, when it comes to applying this regularisation, another problem becomes apparent. As the velocity field is calculated on the grid, so also must the new vorticity field be calculated at the grid node locations. However, if this vorticity field is spread back onto a *random* distribution of particles, new errors are introduced into the particle vorticity field of magnitude similar to those that were just removed, thus invalidating the procedure.

It has been concluded that the only possible means of tackling this divergence problem is to ensure that divergences are not allowed to arise in the first place, as no technique has been found in the context of the current method that removes the singularities without unacceptably altering the structure of the vorticity field.

Close examination of the relative magnitudes of errors introduced in the various calculations involved in the stretching term show that there are two dominant sources of error of similar magnitude. The first error is introduced by the finite difference operator in calculation of the velocity gradient tensor. This error can be reduced by increasing the order of the FD stencil, and/or by reducing the node spacing on the Eulerian mesh. The second error is caused by interpolation

of the gradient tensor onto the particle locations. While this error can also be reduced by minimising the inter-nodal spacing, it should also be noted that this error reduces to zero when the particle location is coincident with the mesh node.

The fact that the interpolation error is completely removed when a particle is exactly coincident with a mesh node suggests using the mesh node locations as the stencil for the regridding step. As shown in Fig. 4.3 this results in a particle distribution of identical density to that of the Eulerian mesh nodes. This is dramatically different from the sort of particle distributions seen in early 2D Particle-in-Cell techniques, where the emphasis was on using a very large 'cloud' of particles on a rather coarse grid.

In order to get a 3D technique to work therefore, it is necessary to have a regular particle location redistribution where a single particle is generated at each grid point, thus removing the large interpolation error just mentioned. Clearly, this limits the particle/cell ratio to unity. To date, the only successful 3D PIC techniques have used this approach, e.g. [32]. During regridding, particle strengths are interpolated onto the mesh using the M_4' interpolation function, and the old particle set is replaced by a new set of particles released from the grid point. Only particles whose strengths are within a certain percentage (0.1% in current study) of the maximum particle strength are created.

An added benefit of this regridding is that it tends to eliminate spurious sub-grid scale structures introduced by distortion of the particle field. This will help to dissipate any developing divergences in the vorticity field, so acting as a sort of implicit divergence filter. A practical requirement of the viscous diffusion algorithm is that there must be 'ghost' particles available to spread vorticity onto from the existing particles, as the viscosity will always tend to increase the support of the vorticity field. So, if the viscous diffusion is applied directly after the regridding, this condition will be guaranteed to be true.

4.6 Error Analysis

The important stages to be examined for the generation of error are those associated with the stretching term calculation. The steps analysed are the interpolation of data from the mesh to the particles and the calculation of spatial gradients using finite difference stencils on the uniform mesh.

The errors are analysed using an ensemble average of 100 runs, each containing 50 sinusoidal waves of random amplitude, phase and frequency. The corresponding Nyquist frequency for the largest mesh spacing analysed is used at the upper limit of frequency, and the overall signal amplitude is normalised to unity after summing all of the signals.

4.6.1 Interpolation

Two different interpolation functions are analysed, the *triangular-shaped cloud* and the M_4' modified B-spline of Monaghan, [33]. The form of these functions is, respectively

$$M_3(\mathbf{x}) = \begin{cases} 0 & \text{if } |x| > \frac{3}{2} \\ \frac{1}{2}(-|x| + \frac{3}{2})^2 & \text{if } \frac{1}{2} \leq |x| \leq \frac{3}{2} \\ \frac{1}{2}(x + \frac{3}{2})^2 - \frac{3}{2}(x + \frac{1}{2})^2 & \text{if } 0 \leq |x| \leq \frac{1}{2} \end{cases} \tag{4.12}$$

$$M_4'(\mathbf{x}) = \begin{cases} 0 & \text{if } |x| > 2 \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 \leq |x| \leq 2 \\ 1 - \frac{5x^2}{2} + \frac{3|x|^3}{2} & \text{if } |x| \leq 1 \end{cases} \tag{4.13}$$

Theory states that the triangular-shaped cloud function should be second-order accurate, with the modified B-spline expected to be third-order accurate. The tests prove this to be so, as shown in Fig. 4.4*(a)*. Fig. 4.4*(b)* shows the spatial distribution of this error. For the M_4' function, this error increases from zero when the measurement point is coincident with a mesh node to a maximum

at a point equidistant between two nodes.

4.6.2 Finite Difference

Two finite-difference stencils are examined, the generic two-point centered difference stencil, and the extended 5-point stencil centered stencil, the development of which is described in A.2.

The results of varying the mesh spacing are graphed in Fig. 4.5. Clearly the higher-order stencil performs much better that the two-point stencil. It is found that the magnitude of this error is approximately double that of the interpolation error using the M_4' function for this particular data set, though the relative magnitudes of these errors are quite sensitive to choice of input data.

It should be noted that this analysis has only been performed for the one-dimensional case. The actual three-dimensional case will correspond to a product of three such errors, clearly making the situation worse.

The truncation error due to the single-point precision of the Poisson solver calculations was also examined. It was found to have a negligible effect on results when compared to the interpolation and quadrature error.

4.6.3 Conclusion

The importance of performing regular regridding, so that the particles are near-coincident with the mesh nodes is made clear from this analysis. Directly after regridding with the M_4' function there is no interpolation error associated with spreading any information between the particles and the mesh.

The conclusion from this brief error analysis must be that, when performing regular regridding of particles onto a distribution which is coincident with the Eulerian mesh nodes, quadrature error from the finite difference calculations on the stretching term become the dominant source of error. This error scales with the cube of the mesh density. This mesh density is extremely limited however due to computer resources, and so this sets the upper limit on accuracy of the

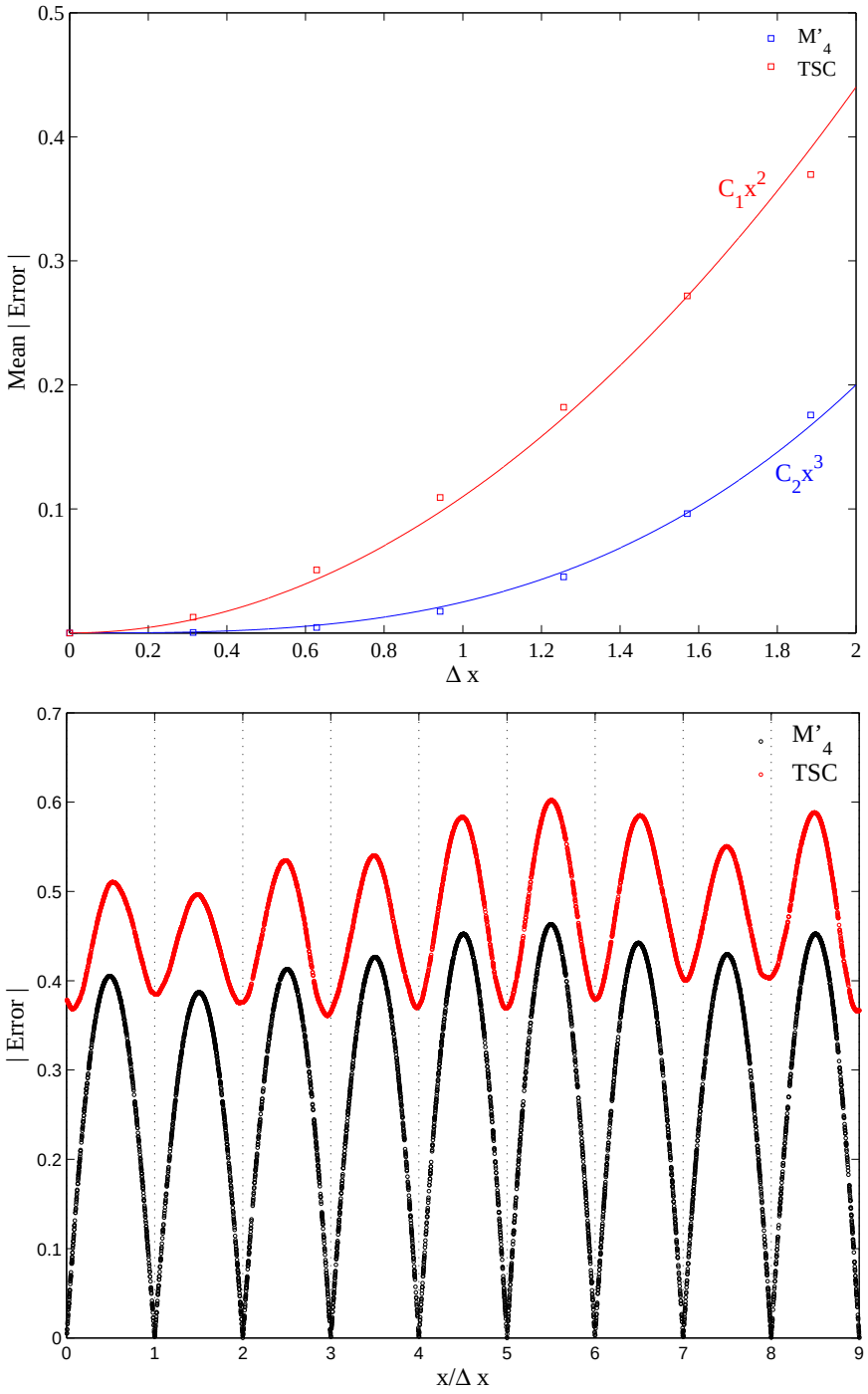


Figure 4.4: *(a)* Magnitude of interpolation error *vs.* node spacing. *(b)* Magnitude of mean error

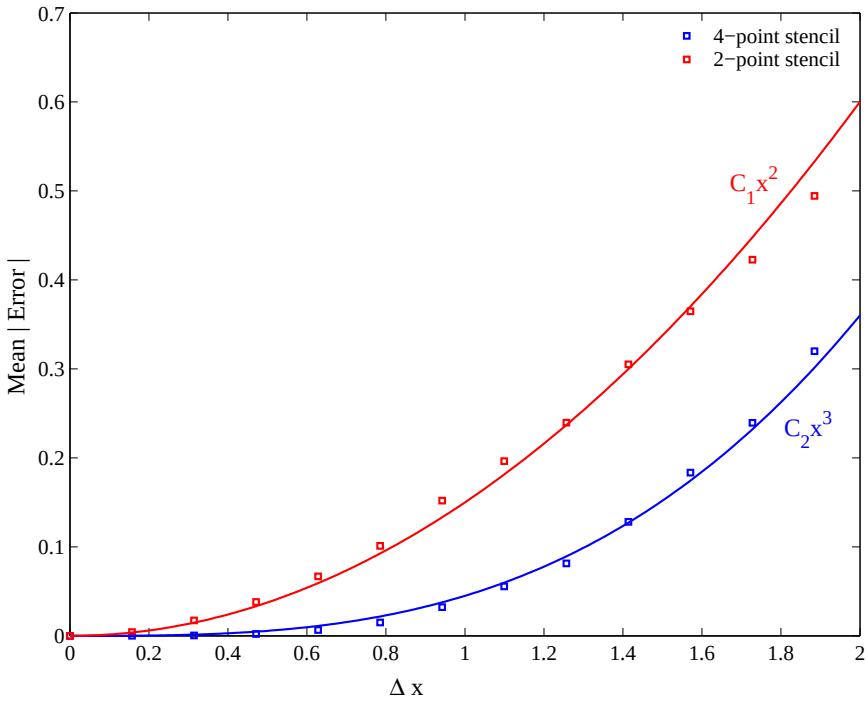


Figure 4.5: *(a)* Magnitude of finite difference error *vs.* node spacing. *(b)* Magnitude of mean error

technique. Not only does this limit spatial accuracy but a certain level of accuracy is required to ensure solution stability, as previously mentioned this error also introduces divergence into the vorticity field.

Chapter 5

Applications

Contents

There are two types of flow used in analysis of the performance of the solver. Vortex rings in various geometries are used initially, as they are easy to initialise and there has been a lot of research, both experimental and numerical, previously carried out on this topic. Same examples are given in Fig. 5.1 and Fig. 5.2. The first of these demonstrates the vortex 'game' first described by Helmholtz (*See § 3.4.1*) in which a pair of coaxial vortex rings of identical circulation leapfrog through each other. Fig. 5.2 demonstrates the oblique collision of a pair of vortex rings which share a single axis of symmetry. Viscous diffusion causes the rings to fuse into a single ring initially, but ultimately a new pair of rings is formed at approximately 90^o to the initial geometry.

The second flow regime studied is that of flow from a circular orifice. If this flow is stopped at a subsequent time, a vortex ring is generated and convects downstream under its self-induced velocity. A experimental demonstration of this

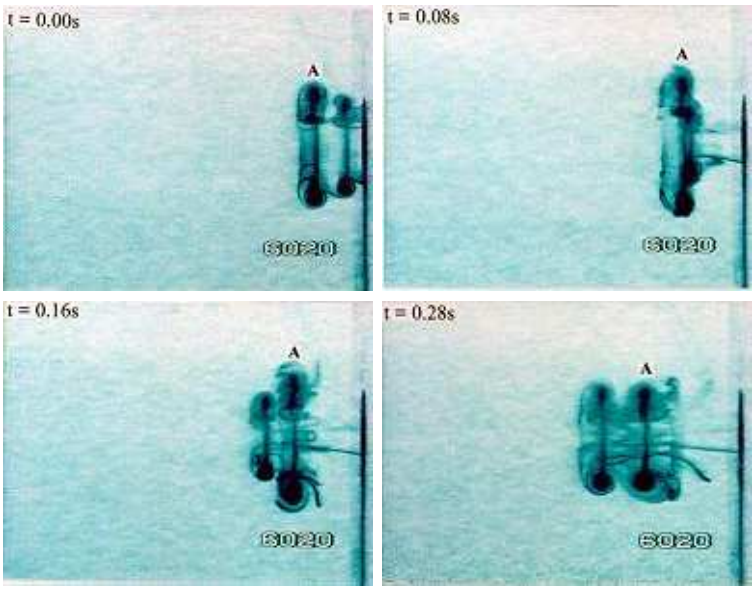


Figure 5.1: Video stills taken from experiment by Lim [1] of leapfrogging vortex rings

behaviour is shown in Fig. 5.3. If fluid is continuously injected into the domain, a free jet develops. The modelling of such a free jet was the initial objective of this work (*See 1.1*) but proved a far more difficult task than the other flows considered. This flow is illustrated in Fig. 5.4, and demonstrates extremely well the main instability mechanism of such a jet, in which the free shear layer breaks up into a series of vortex rings.

§ 5.1 in this chapter deals with the development of a circulation shedding model, which is necessary for the modelling of the jetting flows. § 5.2 briefly discusses the initialisation of vortex rings.

5.1 Circulation shedding model

A numerical model is required to simulate the physical nozzle, which is injecting fluid, and therefore vorticity, into the flow. It is required that this model be capable of generating an arbitrary velocity profile, for example, that produced

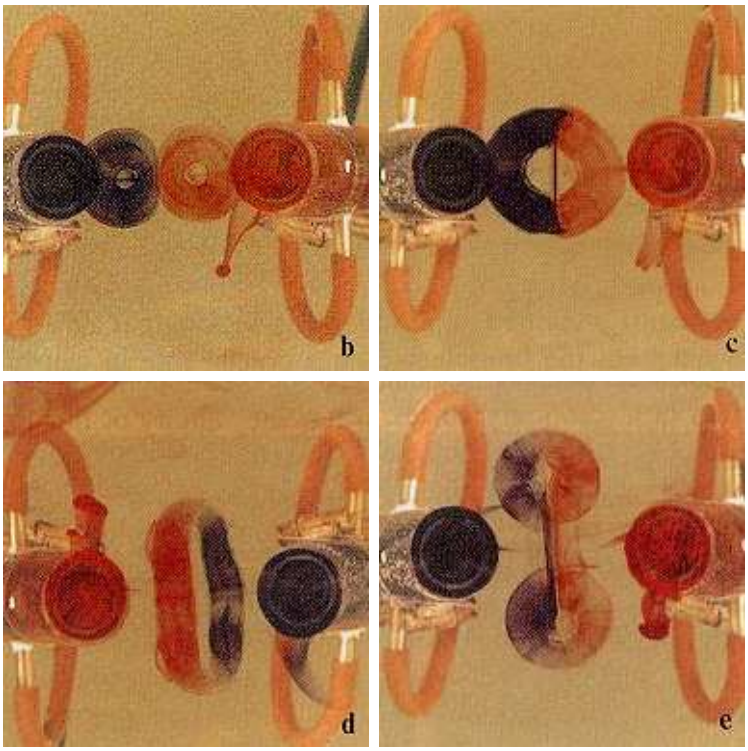


Figure 5.2: Video stills taken from experiment by Lim [2] of obliquely colliding vortex rings

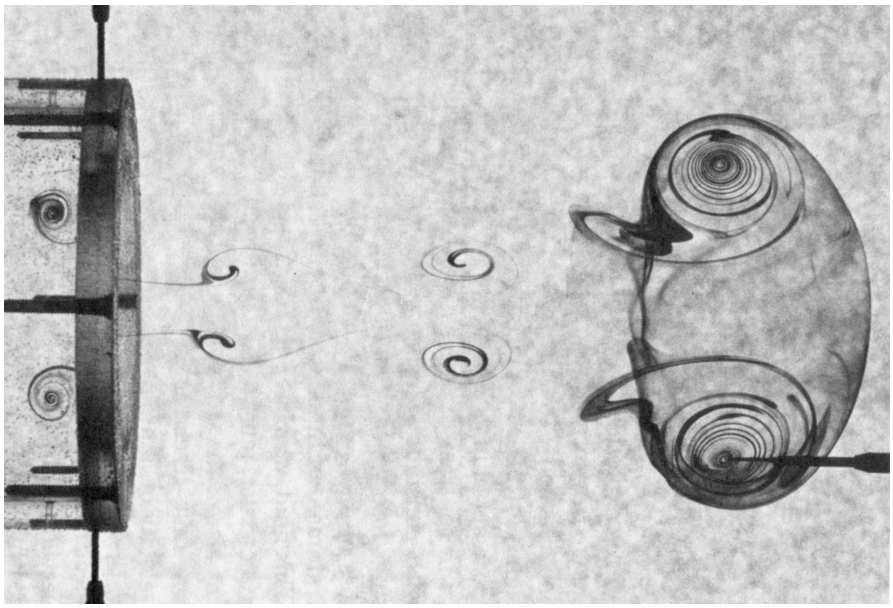


Figure 5.3: Vortex ring formation, visualised by dyed streaklines *(D. Sallet, University of Maryland)*

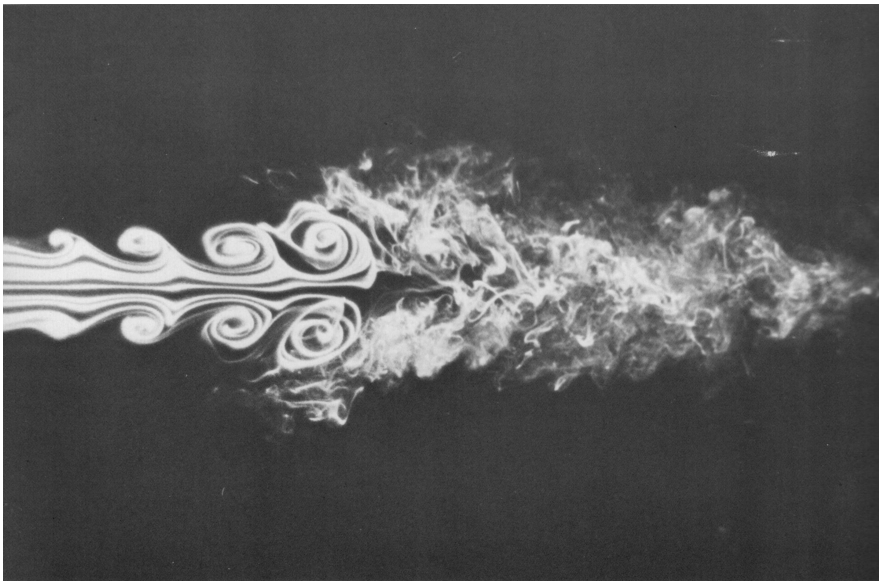


Figure 5.4: Round jet instability: A jet at Reynolds number 10,000 develops axisymmetric oscillations, rolls up into vortex rings, and then abruptly becomes turbulent. *(Drubka & Nagib)*

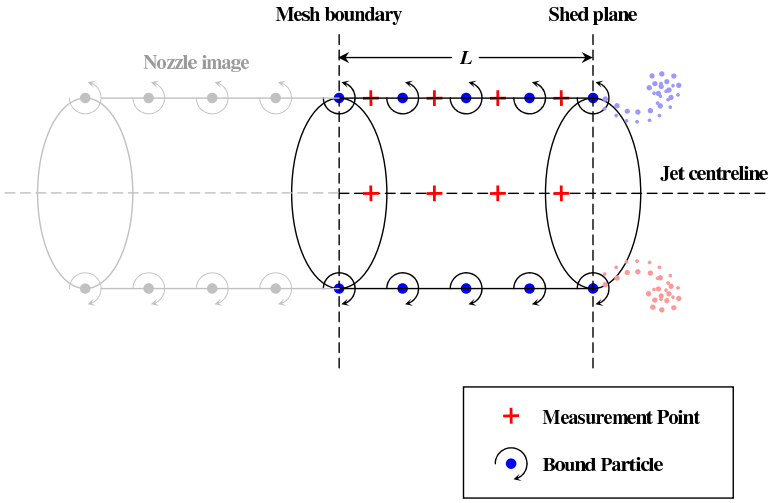


Figure 5.5: Geometry of nozzle vortex-sheet model

by a coaxial jet. Temporal variations in the jet centreline velocity must also be accommodated, to allow for simulation of piston-driven flows generating vortex rings, an example of which is shown in Fig. 5.3. The only limitation placed on the inlet profile is that it be axisymmetric in structure. This restriction, coupled with the assumption of parallel flow inside the nozzle, reduces the model to a one-dimensional problem.

Vortex particle-in-cell techniques have traditionally been regarded as poor at modeling vorticity generation from immersed boundaries orientated parallel to the flow, as is the case here, so a simplified bound vortex model is used to represent the nozzle. A diagram of the various features of this model is given in Fig. 5.5. This approach is based on the technique used by Nitsche and Krasny [34].

The nozzle volume is filled with bound particles. The strengths of these particles are modified at each time step to maintain the required velocity profile at the nozzle exit plane. The particles representing the nozzle wall are shown, as blue circles, in Fig. 5.5.

An influence matrix is solved at each shedding interval to calculate the bound and shed particle strengths. This matrix is generated using an analytical expres-

sion for the velocities induced by a set of singular axisymmetric vortex filaments located at the corresponding radial positions to the bound locations (§.6.1.1). The right hand side of the equation set is defined at a set of measurement points with the following velocity boundary conditions,

- zero-permeability on the wall of the nozzle, $U_r(x, R_\mathrm{j}) = 0$
- specified jet velocity on the nozzle centreline, $U_z(x, 0) = U_\mathrm{j}$.

A polar array of measurement points is used, with 36 azimuthal stations, and the average of the radial velocity at each axial station is used as input to the right hand side. The Eulerian mesh is used to recover the velocities at these points. Marked as red crosses in Fig. 5.5, the measurement points are also staggered axially between the bound point locations. This is in an effort to reduce the condition number of the equation set, as it is quite susceptible to instability, especially at early times. The inflow velocity boundary condition on the Eulerian mesh has the effect of an image nozzle at the measurement locations, and the influence matrix must be modified accordingly.

The equation set solved is shown in Fig. 5.6. A standard least-squares minimisation is used.

For each of the simulations shown, an analytical expression for the inlet velocity profile, given in equation 5.1, is used. This profile is plotted in Fig. 5.7.

$$U_z(r) = U_j \frac{\tanh[(R_\mathrm{j} - r)/\sigma]}{\tanh[R_\mathrm{j}/\sigma]} \qquad \text{for } r \leq R_\mathrm{j} \tag{5.1}$$

Where R_j is the jet nozzle radius, U_j the specified centreline velocity and σ is a parameter controlling the smoothness of the resulting profile. $\sigma = 0.25R_\mathrm{j}$ is used in all simulations, and the resulting profile is plotted in Fig. **??**(a).

This velocity profile is differentiated using the standard finite difference stencil to calculate the required vorticity at the nozzle exit. If the velocity distribution is an analytical expression, as in equation 5.1, the required vorticity can be cal-

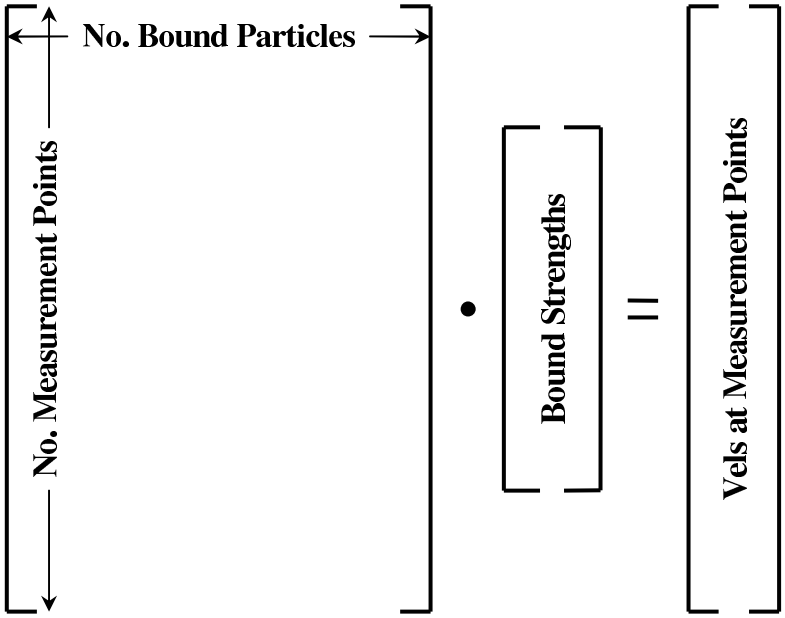


Figure 5.6: Equation set solved using least squares

culated directly. Discretisation of this vorticity onto the particles requires some care, as it can be a large source of error.

The obvious approach is to initialise particles with strengths equal to the specified vorticity at their locations scaled by their associated volumes. However, as the maximum possible mesh density has $\Delta x \approx 0.05 R_{\mathrm{j}}$, there is an unacceptably large discretisation error introduced using this approach. As is evident in Fig. 5.8*(b)*, it gives a poor representation of the required vorticity field. There is a severe 'staircasing' effect at the edge of the nozzle, which can also be seen on inspection of a vorticity isosurface of a developing flow, Fig. 5.10*(a)*.

The effect this error has on the flow is to artificially introduce streamwise vorticity at the locations of maximum error, because errors are introduced into the off-diagonal terms for w in the velocity gradient tensor, $(\frac{\partial w}{\partial x}, \frac{\partial w}{\partial y})$

A means of reducing this error is to subsample within the associated volume of each particle and sum the measured values of vorticity at these locations to give an averaged value to be used for the strength of the particle. This can be thought of as a simple numerical integration, and will give much better agreement for the

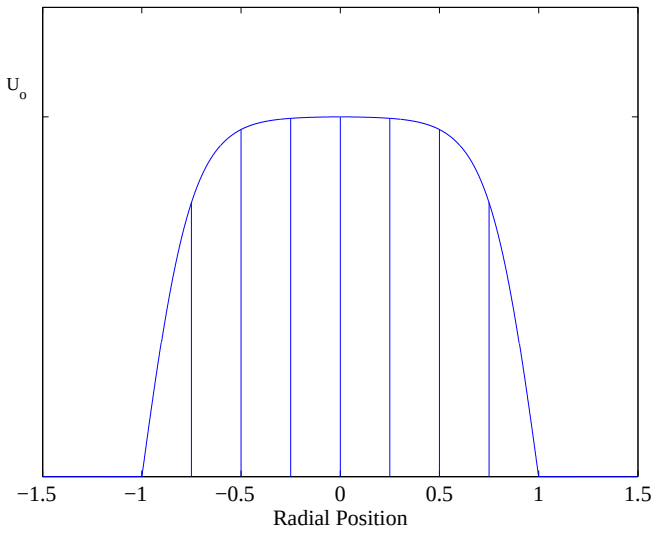
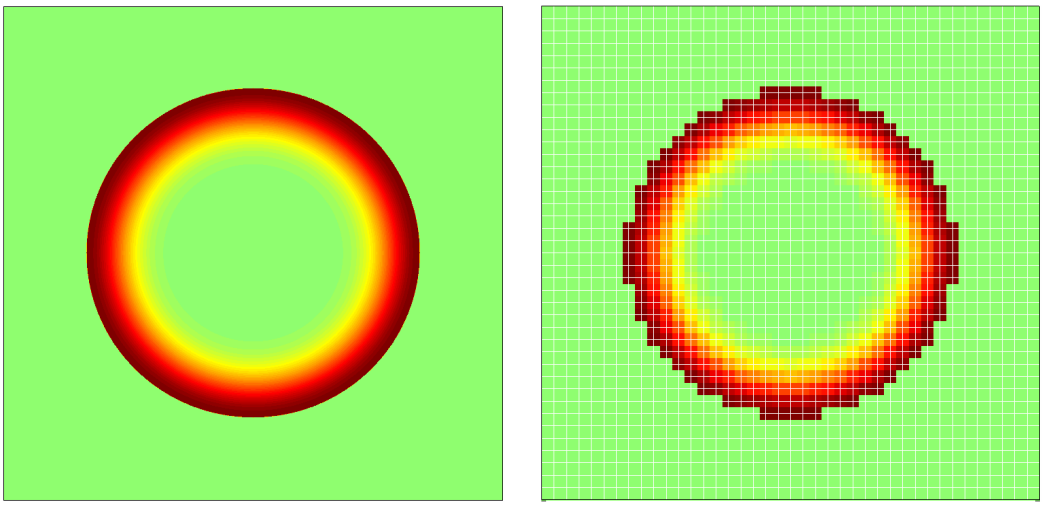


Figure 5.7: Spatial velocity profile, defined in equation 5.1



(a) *(b)*

Figure 5.8: *(a)* Required vorticity profile at jet exit. *(b)* Direct discretisation

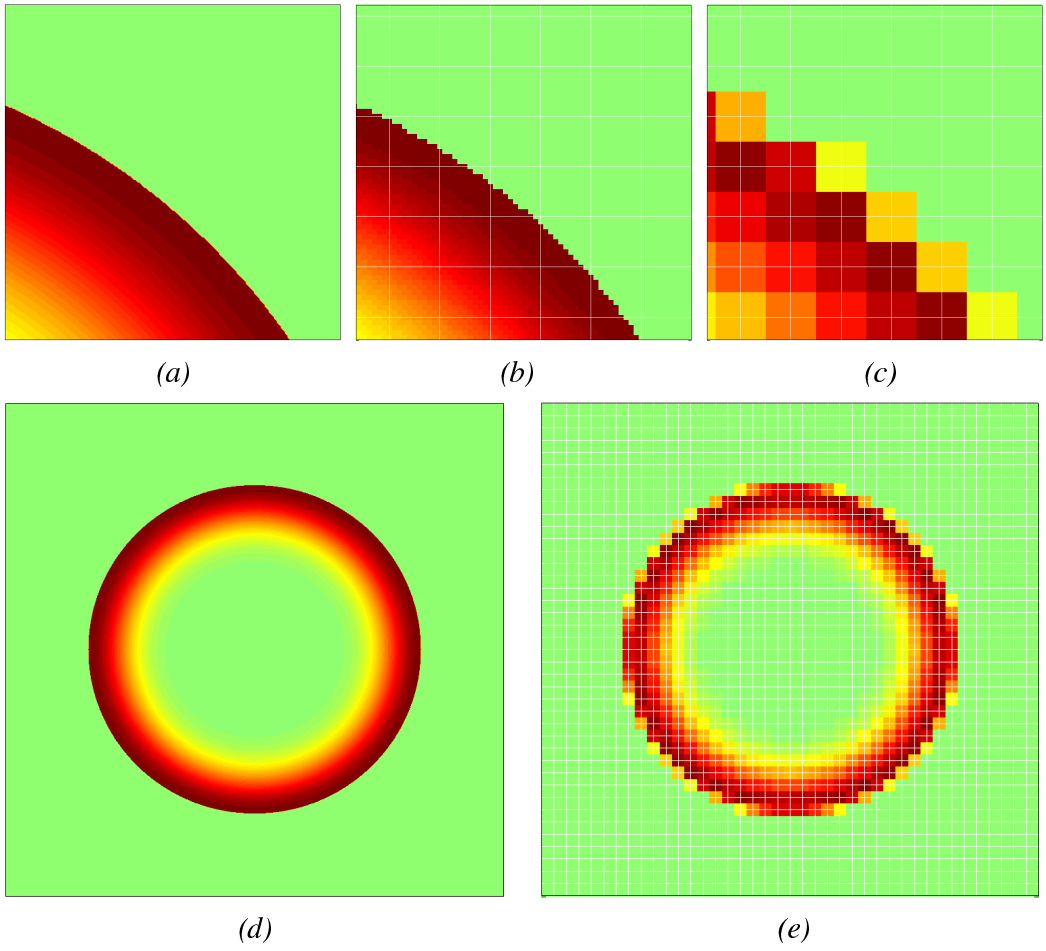
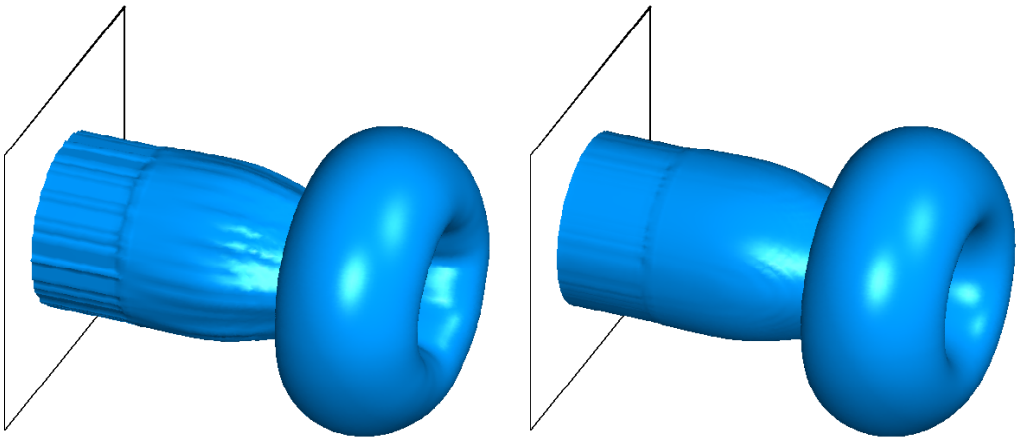


Figure 5.9: *(a)* Section of required vorticity profile at jet exit. *(b)* Subsampled vorticity field. *(c)* Summation of sub-samples at particle location. *(d)* Required vorticity profile at jet exit. *(e)* Discretisation using subsampling

value for total vorticity. A 2D example of this procedure is illustrated in Fig. 5.9. In the actual 3D algorithm, a 100^3 array of measurement points is interrogated within each particle volume, and an average taken of their vorticities.

The end result of this technique, shown in Fig. 5.9*(e)* smooths out the staircasing effect substantially, and for a typical mesh density, reduces it by roughly an order of magnitude. However, it is clear from Fig. 5.10*(b)* that the effect can not be completely removed by this technique. The effects this error has on the flow will be discussed in detail in § 6.6



(a) *(b)*

Figure 5.10: Isosurface of vorticity, *(a)* without and, *(b)* with subsampling at initialisation

At each time step new particles are shed from the jet exit plane. These have an initial vorticity distribution as determined using the subsamping initialisation technique. This distribution is scaled as a function of the particles local slip velocities, in order to satisfy the rate equation for the total shed circulation, Γ_T,

$$\frac{\mathrm{d}\Gamma_T}{\mathrm{d}t} = \int_0^{R_\mathrm{j}} \boldsymbol{\omega}(L,r)\mathbf{u}(L,r)\mathrm{d}r \tag{5.2}$$

Where L is the axial position of the nozzle exit plane. As will be seen in the results section § 6.5, the shed circulation rate is the critical measure of the accuracy of the circulation shedding model, as it can be compared directly to both experimental and theoretical data.

5.1.1 Velocity Perturbation

A random perturbation is applied to the velocity field at the nozzle exit plane. This is in an effort to allow the flow to develop naturally and to 'choose' its instability mode. As described in the previous section, when there is no perturbation applied at this location, it was found that the initial vorticity discretisation artificially excites certain modes at the locations of highest discretisation error. This

gave unnaturally high levels of streamwise vorticity, both at an earlier time than expected, and too far upstream. Gaussian noise is used as the perturbation function. The standard deviation of the perturbation is set at 5% of the characteristic velocity for each of the three velocity components.

5.2 Vortex Ring Initialisation

One of the main advantages of using vortex rings as models for testing the solver performance is that they are quite simple to initialise. The distribution of vorticity within the vortex core is most often assumed to be Gaussian, although other functions have been applied (See [3] for a detailed discussion). There are two standard Gaussian distributions,

$$\boldsymbol{\omega}_\theta = \frac{3\Gamma_o}{2\pi\gamma(\frac{2}{3})\sigma_o^2}\, e^{-\frac{|\mathbf{x}|^3}{\sigma_o^3}}$$

$$\boldsymbol{\omega}_\theta = \frac{\Gamma_o}{\pi\sigma_o^2}\, e^{-\frac{|\mathbf{x}|^2}{\sigma_o^2}}$$

The model in Eq. 5.2 is used in [35], while the second core model is used in various other simulations, for example [36], [37].

This vorticity distribution must be initialised on the cartesian mesh. As computational limitations force the use of a rather coarse mesh, the vorticity in a cell is sub-sampled at the initialisation stage to give a smoother vorticity distribution. This uses an algorithm identical to that used when initialising the shed vorticity in the vorticity generation model (*See § 5.1*).

Application of this initialisation to rings not aligned with one of the major coordinates, as is the case with obliquely colliding rings, is a simple trigonometric transformation.

Chapter 6

Results

Contents

6.1 Vortex rings

The evolution of vortex rings is selected as a case study for the validation of the proposed vortex method. This choice is made for several reasons. The vorticity support in vortex rings is compact, and can be easily initialised. For leapfrogging

or colliding vortex rings, there are localised regions of strong strain, and the accurate conservation of invariants in this situation is quite difficult. Regular regridding to remove particle distortions is a necessity.

A practical advantage of modelling vortex ring interactions where the vorticity support remains relatively compact, as is the case with the obliquely colliding and leapfrogging rings, is that the particles can be easily translated after each time step so that the centre of vorticity is always located at the centre of the mesh, and no particles approach the mesh boundaries. Unfortunately this is not the case with the 'head-on' collision of vortex rings, a very interesting problem for studying the development of azimuthal instability, as the rings undergo large increases in radius during the process.

6.1.1 Induced velocity

An important derivation, used in both vortex ring calculations and jet generation (§.5.1), is the velocity field induced by a circular vortex ring. Using Lamb's definition of the stream-function induced by such a ring, it is derived as follows,

$$\psi = \frac{\Gamma}{2\pi}(\rho_1 + \rho_2)(\mathbb{K}(\lambda) - \mathbb{E}(\lambda)) \tag{6.1}$$

Where

$$\begin{aligned} \rho_1 &= \sqrt{z^2 + (r - R_o)^2 + \sigma_o^2} \\ \rho_2 &= \sqrt{z^2 + (r + R_o)^2 + \sigma_o^2} \end{aligned} \tag{6.2}$$

$$\lambda = \frac{\rho_2 - \rho_1}{\rho_2 + \rho_1} \tag{6.3}$$

$$\begin{aligned} \frac{\partial \psi}{\partial \rho_1} &= \frac{1}{2\pi}\left(\mathbb{K}(\lambda) - \frac{1}{2}(1 + \frac{\rho_2}{\rho_1})\mathbb{E}(\lambda)\right) \\ \frac{\partial \psi}{\partial \rho_2} &= \frac{1}{2\pi}\left(\mathbb{K}(\lambda) - \frac{1}{2}(1 + \frac{\rho_1}{\rho_2})\mathbb{E}(\lambda)\right) \end{aligned} \tag{6.4}$$

$$\begin{aligned}\frac{\partial\psi}{\partial z} &= \frac{z}{\rho_1}\frac{\partial\psi}{\partial\rho_1} + \frac{z}{\rho_2}\frac{\partial\psi}{\partial\rho_2} \\ \frac{\partial\psi}{\partial r} &= \frac{(r-R_o)}{\rho_1}\frac{\partial\psi}{\partial\rho_1} + \frac{(r+R_o)}{\rho_2}\frac{\partial\psi}{\partial\rho_2}\end{aligned} \tag{6.5}$$

$$\begin{aligned}u_r &= -\frac{\Gamma}{r}\frac{\partial\psi}{\partial z} \\ u_z &= \frac{\Gamma}{r}\frac{\partial\psi}{\partial r}\end{aligned} \tag{6.6}$$

and $\mathbb{K}(\lambda)$ and $\mathbb{E}(\lambda)$ are the complete elliptic integrals of the first and second kind of λ, respectively.

On the axis, $(r = 0)$, only u_z is non-zero,

$$u_z = \frac{\Gamma}{2}\frac{R_o^2}{\left(z^2 + R_o^2 + \sigma_o^2\right)^{3/2}} \tag{6.7}$$

For a vortex ring with infinitesimally small radius, σ_o in these expressions is set to zero.

6.1.2 Free Convection

The free convection of an unperturbed vortex ring is a useful benchmark for the Poisson solver. Analytical solutions for the self-induced velocities are derived in Saffman, [25]. The self-induced velocity is given by,

$$U = \frac{\Gamma}{4\pi R}\left[\log\frac{8R}{\sigma} - 0.558\right] \tag{6.8}$$

It is illustrated in Fig. 6.1

6.2 Vortex ring instability

This section is concerned with the instability of vortex rings to azimuthal bending waves. These are classed as *short-wave* instabilities, i.e. the wavelength of the perturbation is comparable to the vortex core size.

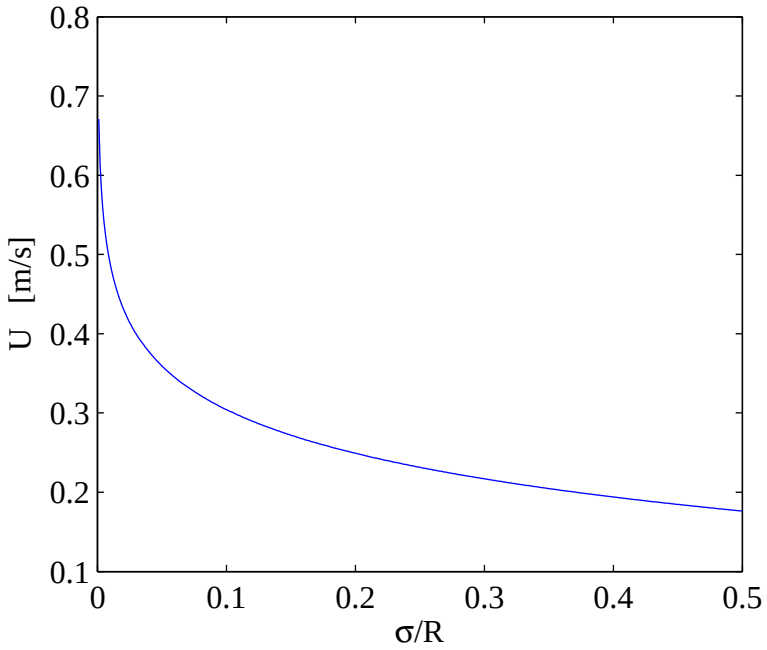


FIGURE 6.1: Analytical solution for self-induced velocity of vortex ring

Core Radius, σ	0.1	0.15	0.2	0.25
Normalised Velocity, $\tilde{V}$	3.824027	3.418562	3.130879	2.907736

TABLE 6.1: Normalised self-induced velocities for varying core radii, $R = 1$, $\Gamma = 1$

A large amount of theoretical, experimental and numerical work has been carried out in investigating this problem.

When a vortex ring is perturbed sinusoidally in the radial direction around its azimuth, the perturbations will reorient themselves along the streamwise direction and, given the proper wave-number, will grow inviscidly in that direction. If the algorithm used is too diffusive, streamwise vortices never develop and the vortex ring remains laminar.

An example from literature uses $\sigma/R = 0.45$–a very 'fat' ring. The vortex core distribution is initially a cubic Gaussian and the ring is perturbed sinusoidally around its azimuth with the most unstable wave number (=7) and 5%

perturbation amplitude.

The wavenumber, k, is the number of waves that are fitted around the entire circumference of the vortex ring. The perturbation amplitude, A, is the magnitude of the perturbation as a fraction of the original radius. A perturbation is applied to an initially circular ring in Fig. 6.2, giving a new expression for the radius of the filament,

Stability theory dictates that some wavenumbers will cause the ring to become unstable, while the effects of others will eventually disappear.

subsectionInitial perturbation

There are two approaches possible to initialise the perturbed vorticity field. These can be termed the *geometric* and *zero-divergence* methods. In the first, the ring geometry is modified, and Helmholtz's law is used to reorientate the vorticity vectors accordingly. In the second, the azimuthal vorticity is initialised in a manner similar to the first, but the radial vorticity is used to explicitly enforce zero divergence in the vorticity field.

$$R(\theta) = R_R + A \cdot sin(k\theta) \tag{6.9}$$

Vorticity vectors are re-orientated to obey Helmholtz's law.

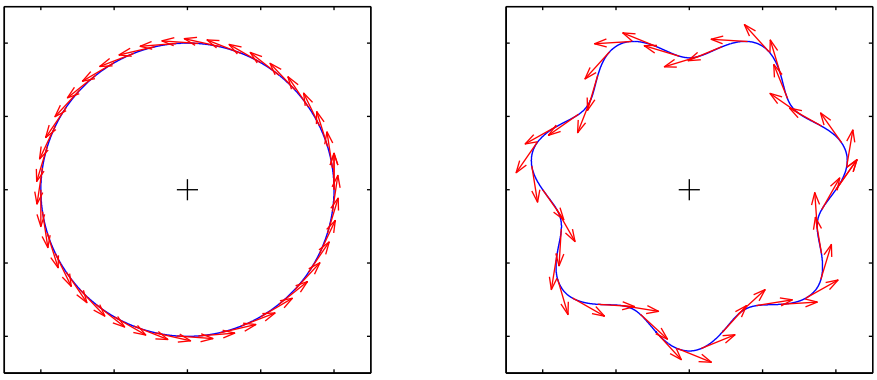


FIGURE 6.2: Application of azimuthal perturbation: $k = 7$, $A = 0.1$. Vorticity vectors shown in red

Fig. 6.3 demonstrates the error incurred in applying a perturbation to the vortex ring. Vorticity divergence is introduced

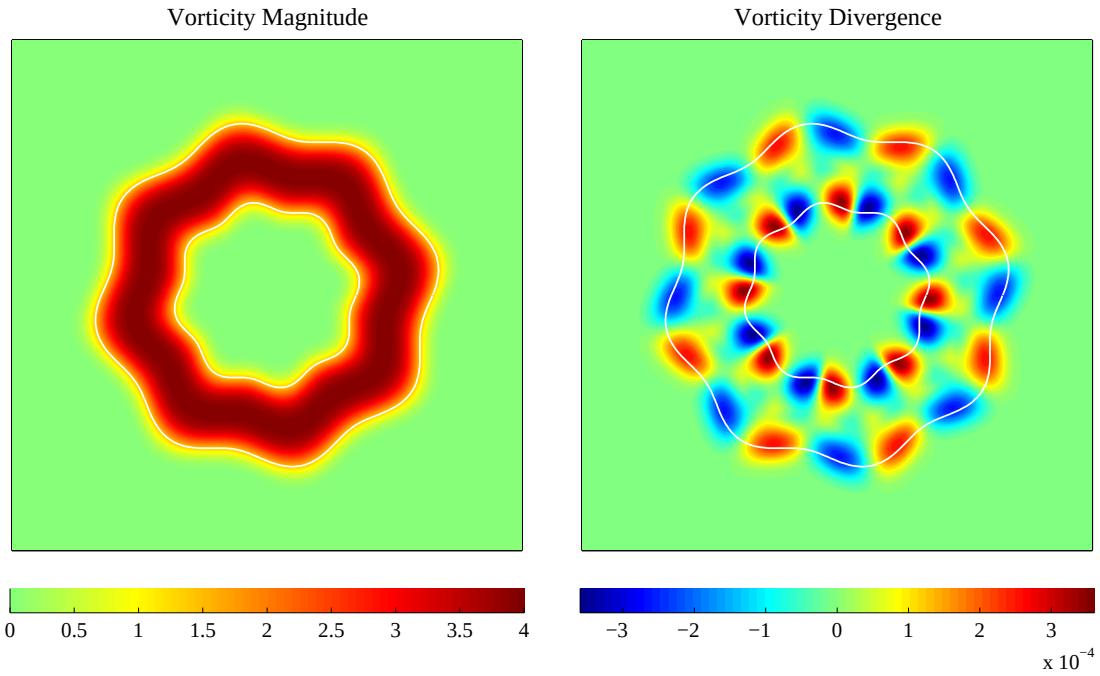


FIGURE 6.3: Initialisation divergence error: $k = 8$, $A = 0.05$, $\Delta = 0.05$

6.3 Leapfrogging rings

6.4 Colliding rings

In this simulation, two rings of equal circulation that share an axis of symmetry are orientated such that they will collide obliquely due to the induced velocity field. This is an interesting problem for several reasons. The first is that it is a completely three-dimensional phenomenon, (*i.e.* the entire velocity gradient tensor is populated). This tests the ability of the stretching term model to accurately reorientate the particle strength vectors, without introducing divergence into the vorticity field. The second interesting feature is what occurs to the rings during collision. Opposite sign regions of vorticity are forced together at a contact 'patch'. Viscous diffusion causes the patch to diffuse totally, generating a single new ring where the free ends fuse together. There is also intense vortex stretching at this point, evident from the large peaks in velocity gradient at this location.

This problem has been examined extensively in the laboratory, notably by Lim [2]. Using extremely precise control of the ring trajectories, the two initial rings were first observed to fuse and, at a later stage, a fission process causes a new pair of rings to form, orientated at approximately 90^o to the first. Some stills taken from a video of this experiment are shown in Fig. 5.2.

An important numerical study of this phenomenon was performed by Winckelmans & Leonard [29][3]. In this work, a purely Lagrangian vortex method is developed, where a set of regularised vortex blobs is used to represent the vortex ring structure. The velocities induced by the particles on each other is calculated using direct summation, and the expression derived in § 4.2 was used to calculate the stretching of the particle strengths. The particle strength exchange technique was used to model viscous diffusion.

The problem geometry is illustrated in Fig. 6.4, which shows a 2D slice through the rings common axis of symmetry. The initial conditions were chosen so as to

exactly replicate the simulation in [3]. These were; Ring major radius $R = 1$, Ring circulation $\Gamma = 1$, Initial separation $S = 2.7$, angle of approach $\alpha_{\mathrm{R}} = 15^o$, Ring core radius $\sigma = 0.2$ and Reynolds number (defined as Γ/ν) of 400.

The velocity field was solved on a 256^3 periodic box, with a uniform mesh spacing of 0.05. A constant time step of $0.01s$ was used. Regridding was performed at every time step.

The strength vectors of the vortex particles from the Winckelmans & Leonard study are plotted in the top series of images. The nearest equivalent in this study is an isosurface of vorticity magnitude, which is shown in the middle series of images. The value of the vorticity isosurface is 10% of the initial peak vorticity magnitude. The bottom series of images show the temporal development of a set of inert scalar particles, which were initialised inside the initial vorticity support at time $t = 0$.

Very good qualitative agreement is demonstrated by these figures, with fusion occurring in the identical manner to the compared work. If the model is allowed to run beyond $8.0s$, which is the furthest the cited study runs, it is possible to see the beginning of the fission process, which was shown very clearly in the experimental work of [2], and also numerically modelled by Cottet [28]. Unfortunately, the Reynolds number used in the current simulation leads to excessive diffusion for this process to complete, the rings 'stagnate', so there is still a bridge of vorticity connecting the two rings. This process is shown at $16.0s$ and $32.0s$ in Fig. 6.8

Comparison of the resulting enstrophy with that obtained in [3] is shown in Fig. 6.9.

Fig. 6.10 plots the number of particles in the domain during the simulation. It is interesting to note that this count decreases initially due to the collision process, but as soon as fusion occurs, the count increases in a manner consistent with the spreading effect due to the viscous diffusion. This count will decrease gradually again at later times due to particles falling below the minimum allowable particle strength, set at 0.05% of the maximum initial particle strength.

6.5 Vortex Ring Generation

The circulation generation model (§ 5.1) is tested for a flow condition matching that in Didden's experiment [4] and the simulation of that experiment by Nitsche and Krasny [34]. A moving piston forces fluid from a circular tube, causing the generation of vortex ring. The piston is accelerated from rest until a time t_u and then translated at a constant velocity until a time t_{off}, when it is impulsively stopped. A schematic of Didden's experiment is shown in Fig. 6.11. The corresponding model diagram was shown in Fig. 5.5. This flow was chosen in order to test fully the performance of the bound vortex model of the nozzle. Fig. 6.12 graphs the ramped start and the impulsive stop of the simulated flow.

$$V_j(t) = \begin{cases} V_o(1 - (\frac{t_u - t}{t_u})^{2.8} & 0 \leq t \leq t_u \\ V_o & t_u \leq t \leq t_{off} \\ 0 & t > t_{off} \end{cases} \tag{6.10}$$

Where $t_u = 0.55$, $t_{off} = 3.0$. All values have been normalized by jet radius and velocity. Both the ramped start and the impulsive stop require very accurate calculation of the circulation shedding rate, and the vorticity gradients generated at these times are much higher than for steady-state jet flows.

Fig. 6.13 plots the vortex ring diameter D versus axial distance x_c for the nozzle exit plane. The values for the current study are taken as the position of maximum vorticity. The values from Nitsche & Krasny's work correspond to three different values of vortex sheet smoothing function, namely $\delta = 0.1$, $\delta = 0.2$ and $\delta = 0.4$.

The ring diameter increases for $t < t_{off}$ as the piston moves and reaches a peak at the instant the piston is stopped. The diameter then decreases for approximately half of the piston 'on' time but begins to increase again after that. A possible explanation for this, given in Nitsche & Krasny, is that it could be caused by 'tumbling' in the core vorticity distribution. Indeed, it is clear

from the vorticity contour plots that the vorticity structure in the generated ring is distorted by the reversing flow at the nozzle exit plane for $t > t_{off}$ so this distortion could lead to a non-constant ring diameter. Once this distortion has been fully absorbed, the ring diameter should decrease roughly linearly due to viscous diffusion.

The data in Fig. 6.13 is evaluated by location of the particle with the maximum vorticity magnitude, and so is quite 'blocky', due to the rather coarse spatial discretisation. The results for the resulting ring diameter and downstream distance travelled shown reasonable convergence with both Didden's experimental data and Nitsche & Krasny's work. However, the current study clearly underestimates both the ring peak ring diameter, and the distance downstream at which this occurs.

There are some important differences between this work and that of Nitsche & Krasny that limit the possible agreement of the results, and may help to explain the clear differences between the two. The vortex sheet used in Nitsche & Krasny is inviscid, whereas in this work there is an artificial boundary layer imposed on the nozzle exit velocity profile (*See Fig. 5.7*) and there is also a viscous diffusion model applied on the shed vorticity. Both of these factors will obviously affect the resulting vorticity field. The values chosen for σ, the parameter controlling the smoothness of the nozzle exit velocity profile (in equation 5.1, and ν, the kinematic viscosity, were both the minimum allowable to ensure solution convergence.

Not having a 'top-hat' velocity profile means that the shed vorticity is dispersed in the radial direction, while there is a singular peak of shed vorticity using the inviscid vortex sheet model. It seems that this inhibits the roll-up of the shed vorticity, and may explain the underestimation of the resulting ring diameter. Viscous diffusion will obviously tend to decelerate the developing ring, so the downstream distance travelled will never be as great as that of an inviscid model.

Fig. 6.14 plots the circulation shedding rate for the vortex ring generation problem. Results from Didden and Nitsche & Krasny are shown as are results for the slug-flow model. This value is calculated in the experimental and the current study using equation 6.5.

$$\frac{\mathrm{d}\Gamma_T}{\mathrm{d}t} = \int_0^{R_j} \boldsymbol{\omega}(L,r)\,\mathbf{u}(L,r)\mathrm{d}r \tag{6.11}$$

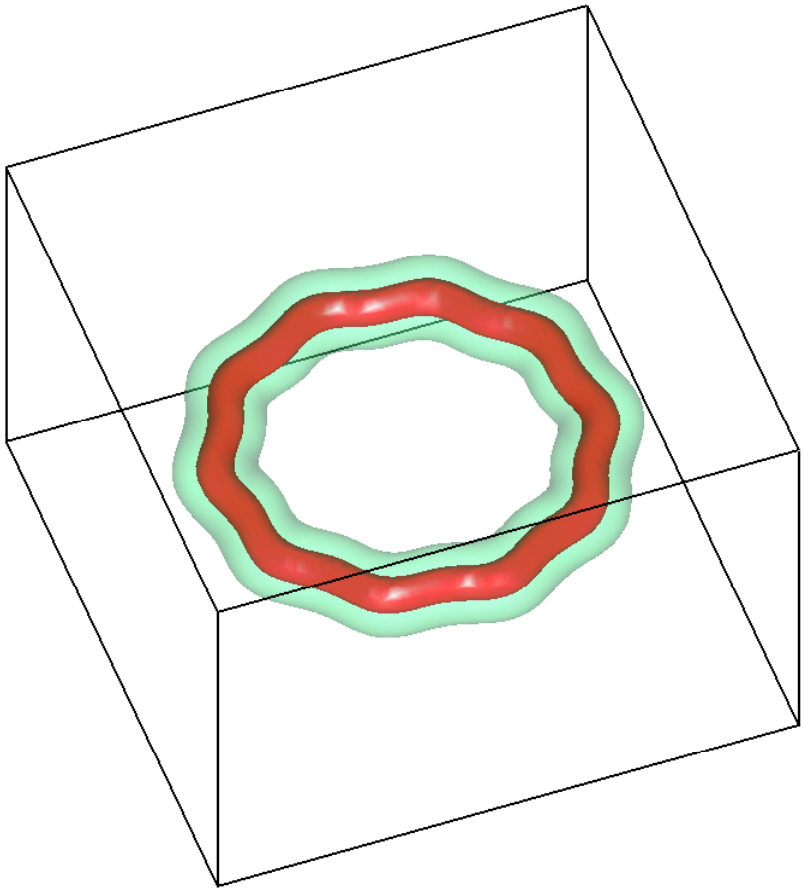
where L is the axial position of the nozzle exit plane. This value is directly available in the vortex sheet model, and is defined as $\mathrm{d}\Gamma_T/\mathrm{d}t = \frac{1}{2}V_j^2$ for the slug-flow model.

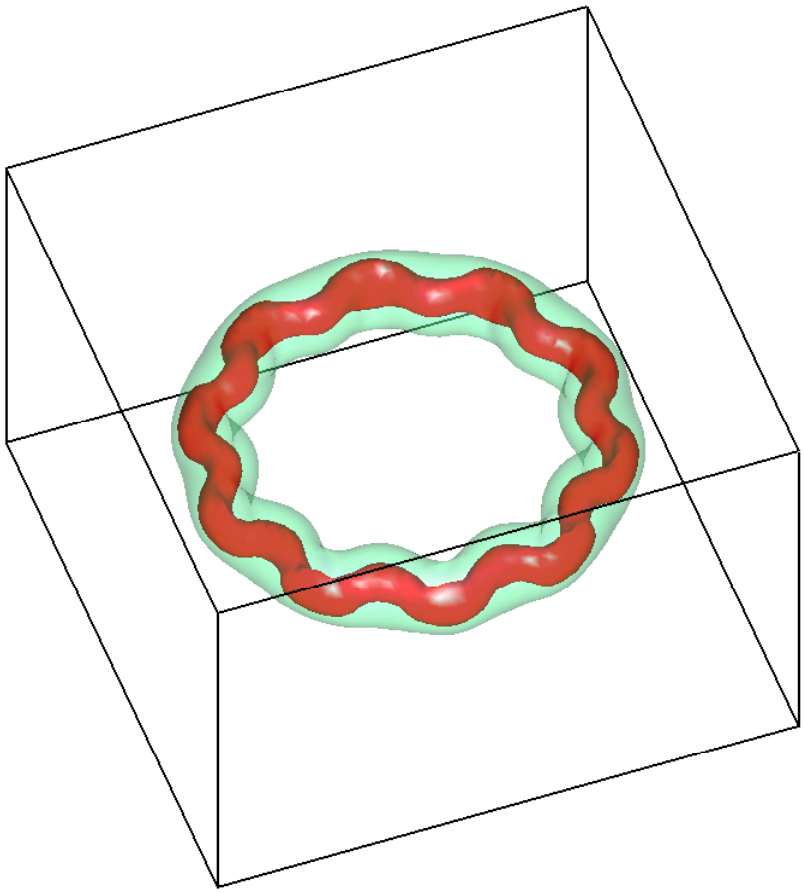
6.6 Free Jets

Fig. 6.18 plots the measured linear impulse for the free jet simulation versus that predicted by theory. The measured linear impulse is calculated using the discretised version of the definition of linear impulse given in equation 3.3. The predicted value is calculated from the external force imposed by fluid injection at the nozzle, defined (per unit density) as $piR_\mathrm{j}^2U_\mathrm{j}^2$.

At approximately $t = 16.0$ particles reach the outflow boundary, and the decay model is applied. The initial dramatic reduction in impulse is not physical, and demonstrates that the decay model has difficulty convecting the large starting vortex out of the domain without affecting the upstream vorticity field. However the linear impulse seems to converge to a steady solution, as expected, after the starting vortex has completely convected out of the system.

Apart from initial overestimation of the linear impulse, there is very satisfactory agreement between theory and the model for the overall linear impulse. This is not unexpected however, as errors introduced from vortex stretching will be much less than for the obliquely colliding rings case, in which there was rather poor conservation of linear impulse.





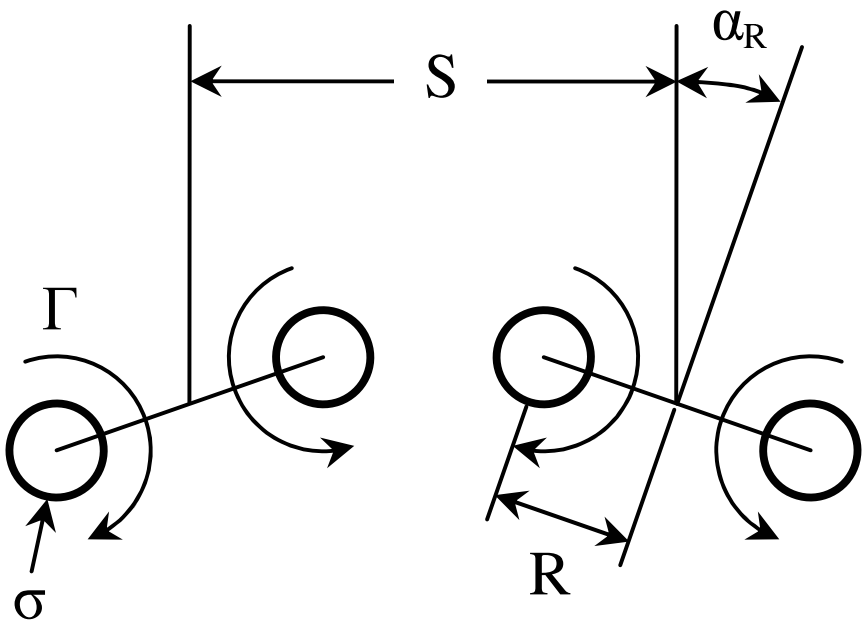


Figure 6.4: Problem geometry for oblique collision of vortex rings

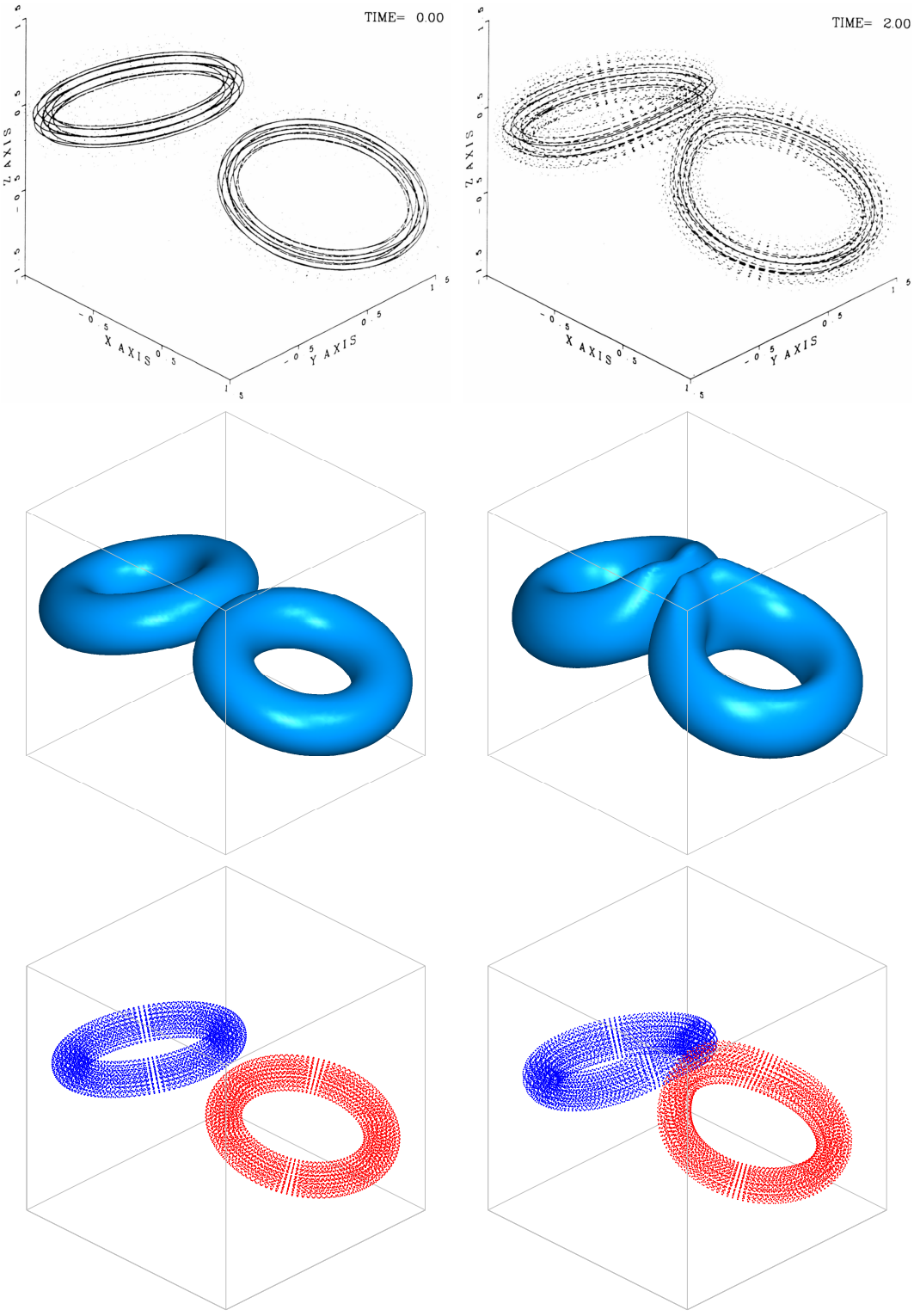


Figure 6.5: Obliquely colliding rings problem, $\Gamma = 1$, $R = 1$, $S = 2.7$, $\alpha = 15^o$. *top* Winckelmans & Leonard simulation. *middle* Vorticity isosurface, $|\boldsymbol{\omega}| = 0.5$. *bottom* Scalar particles.

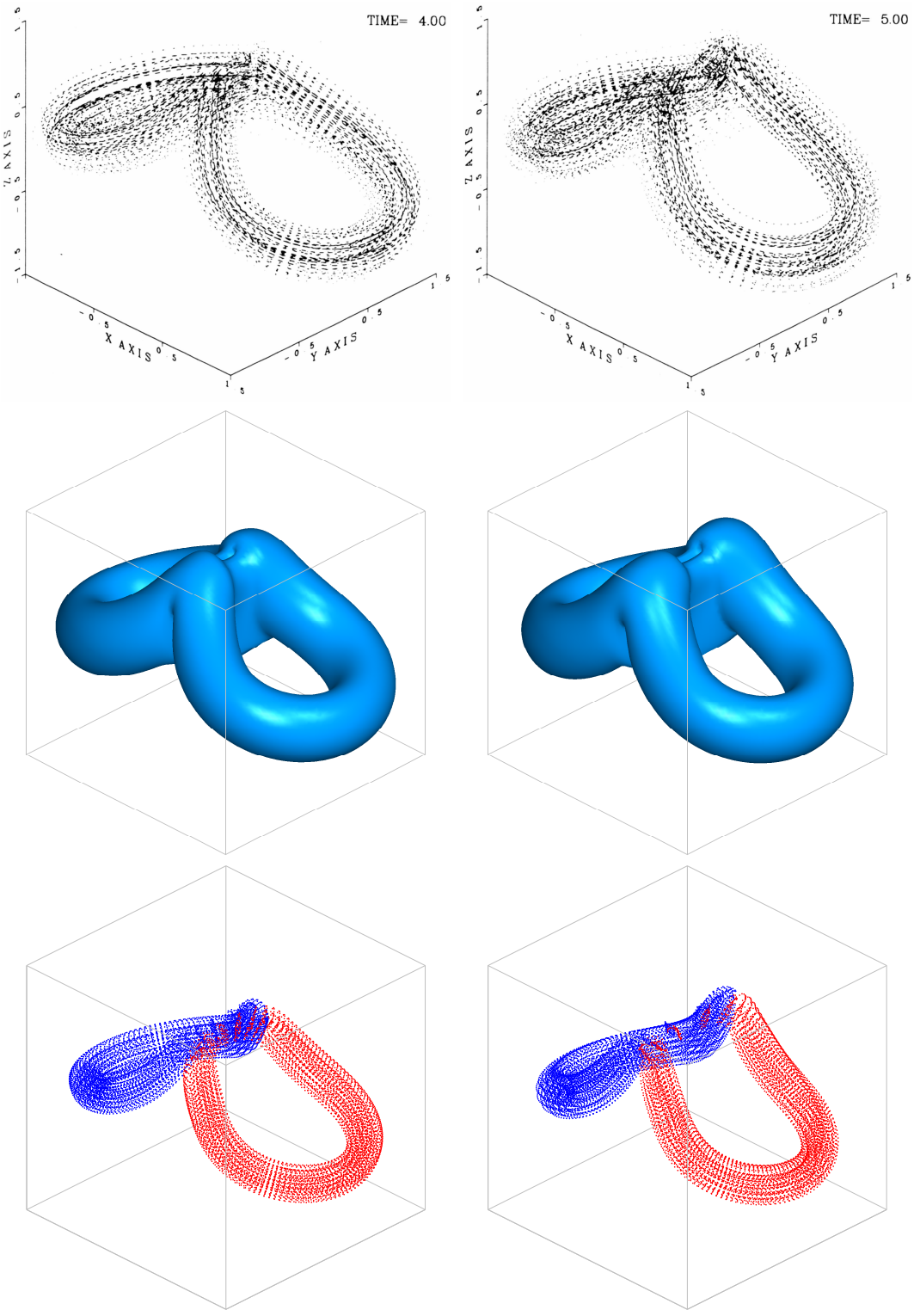


Figure 6.6: Fig. 6.5 *cont.*

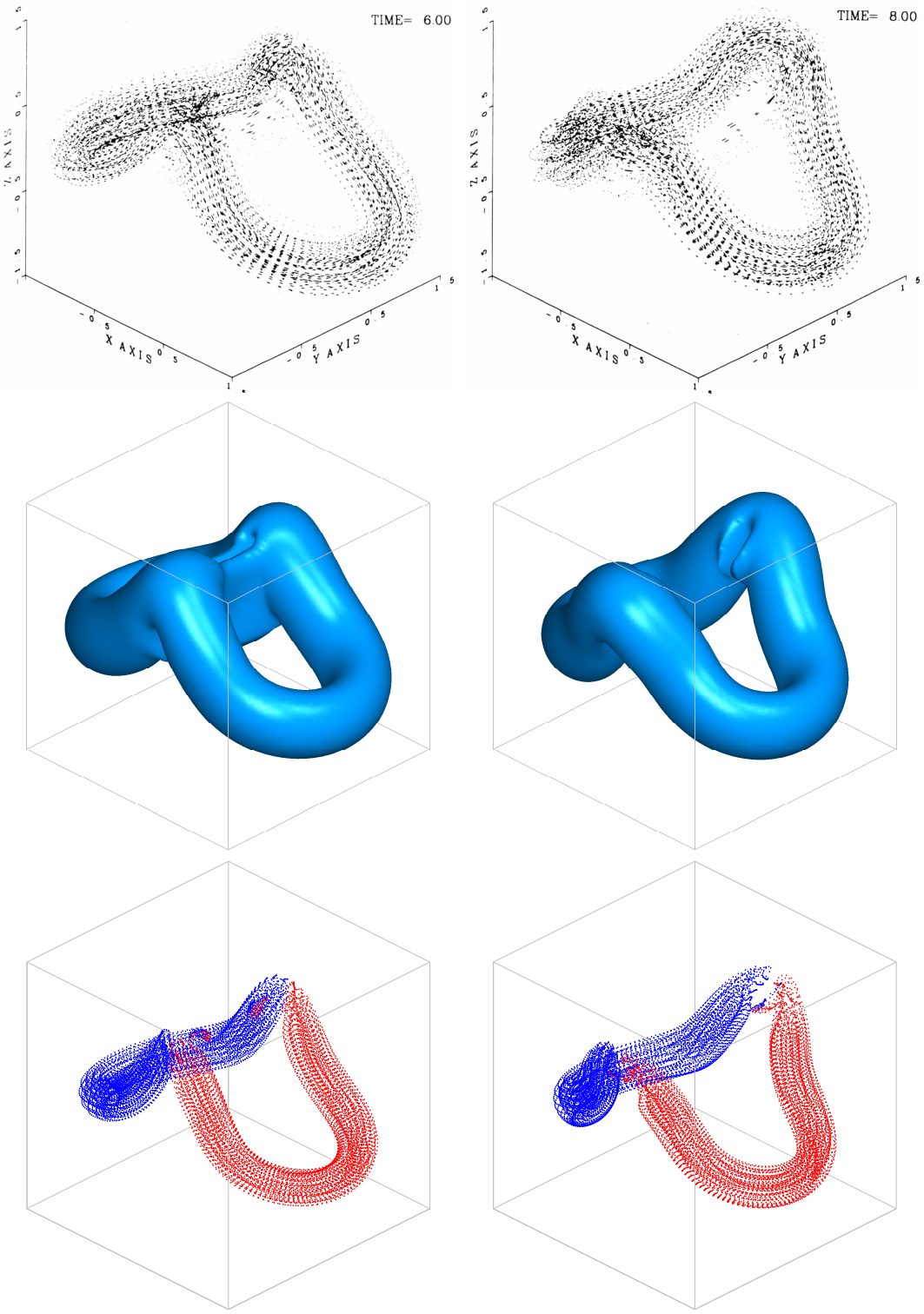


Figure 6.7: Fig. 6.5 *cont.*

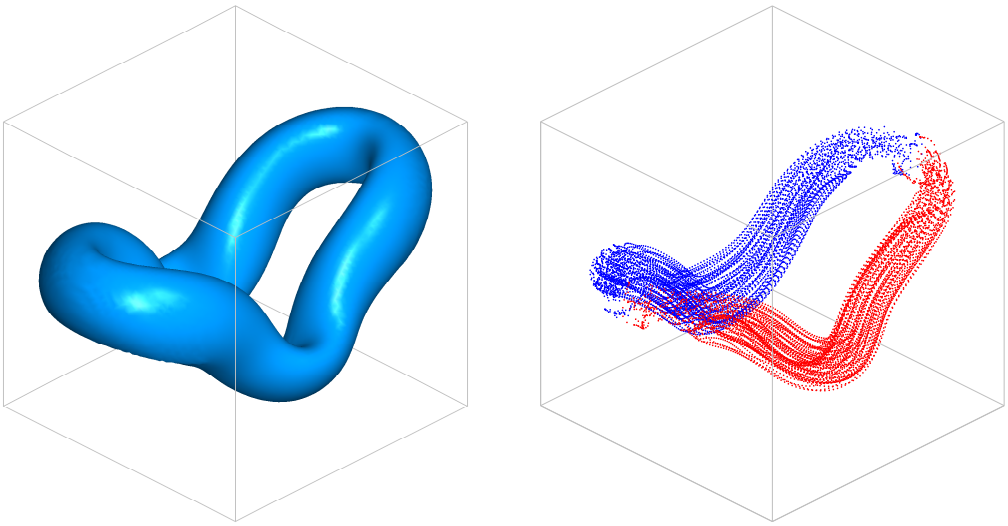


Figure 6.8: Results from current study at T = 16.0

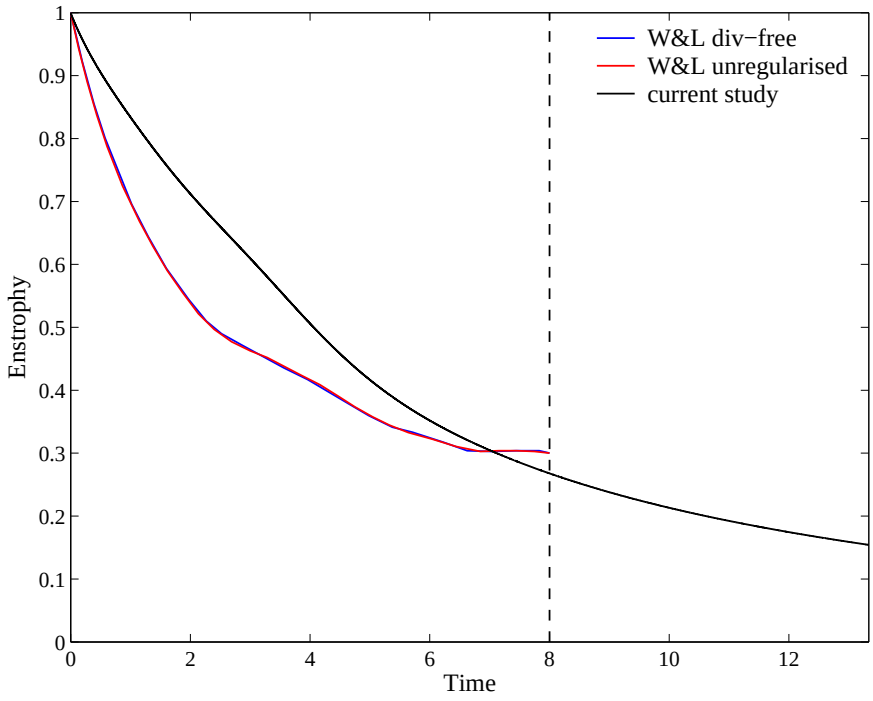


Figure 6.9: Comparison of enstrophy development with Winckelmans & Leonard [3]

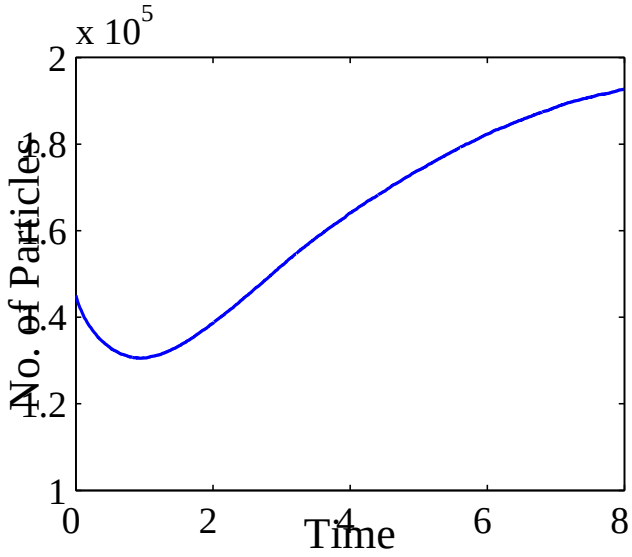


Figure 6.10: Number of particles in domain versus time. Minimum allowable strength $= 5 \times 10^4 \cdot |\boldsymbol{\alpha}|_{max}$ at $t = 0$

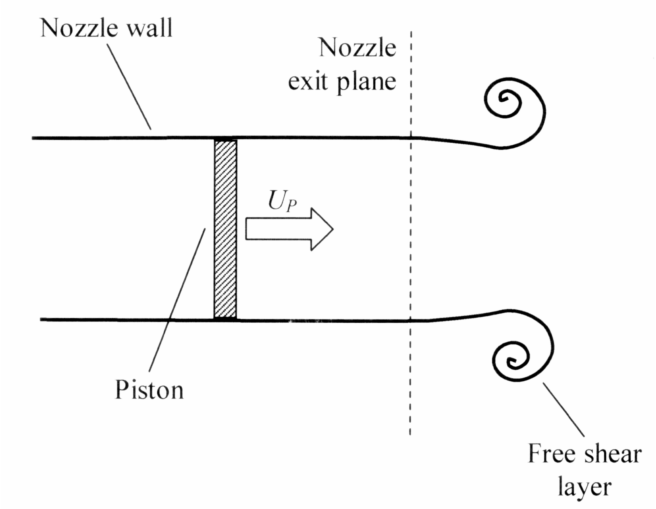


Figure 6.11: Diagram of Didden's experiment [4], showing relevant terminology

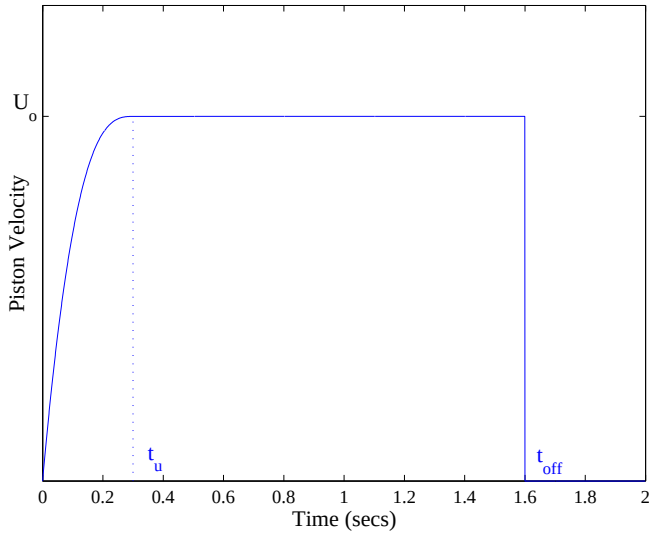


Figure 6.12: Temporal velocity profiles, defined by equation 6.10

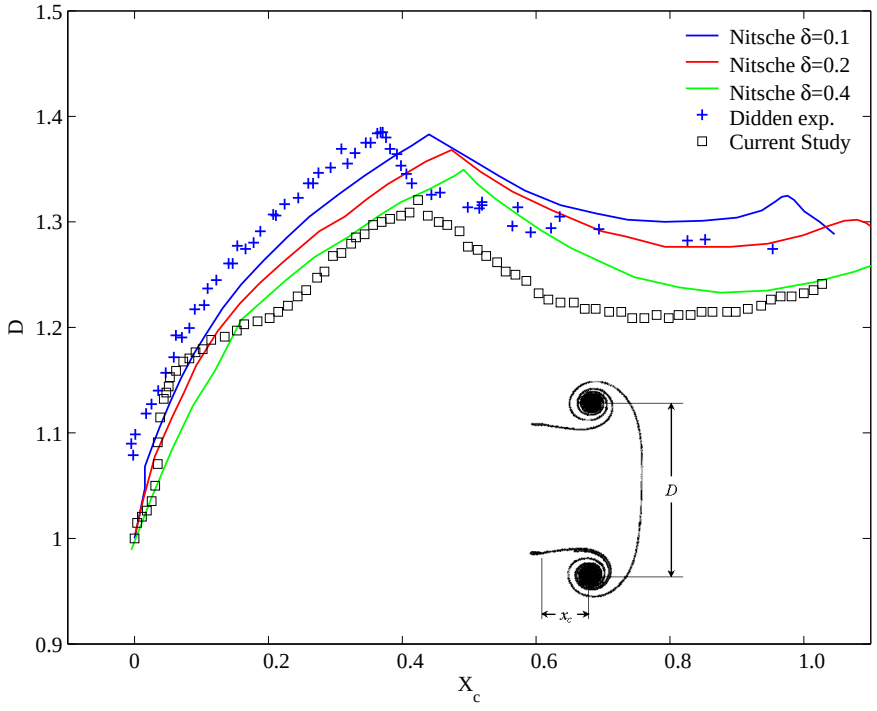


Figure 6.13: Vortex ring diameter *vs.* axial distance from tube edge.

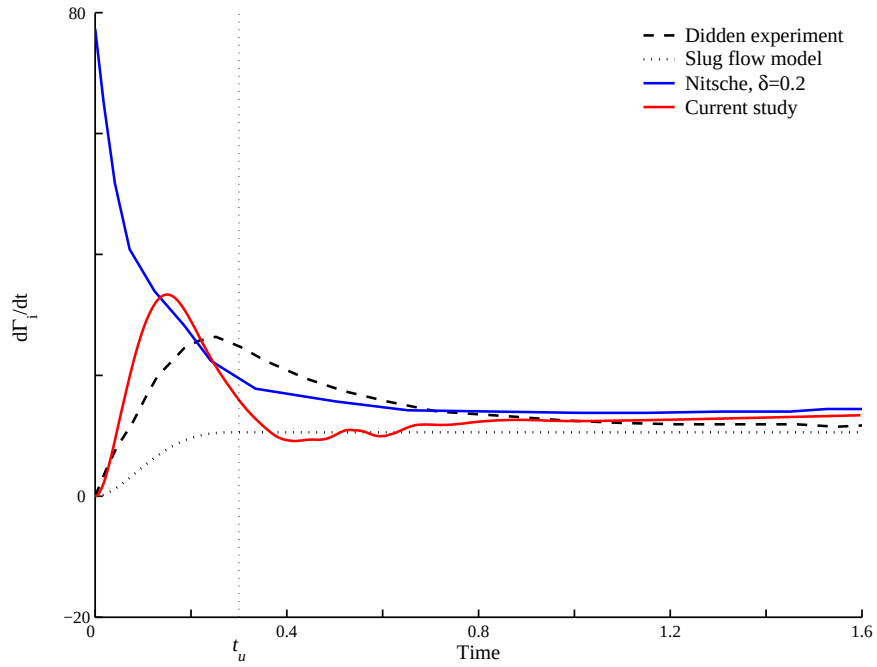


Figure 6.14: Comparison of circulation shedding rate with results from Nitsche & Krasny, Didden and slug flow model

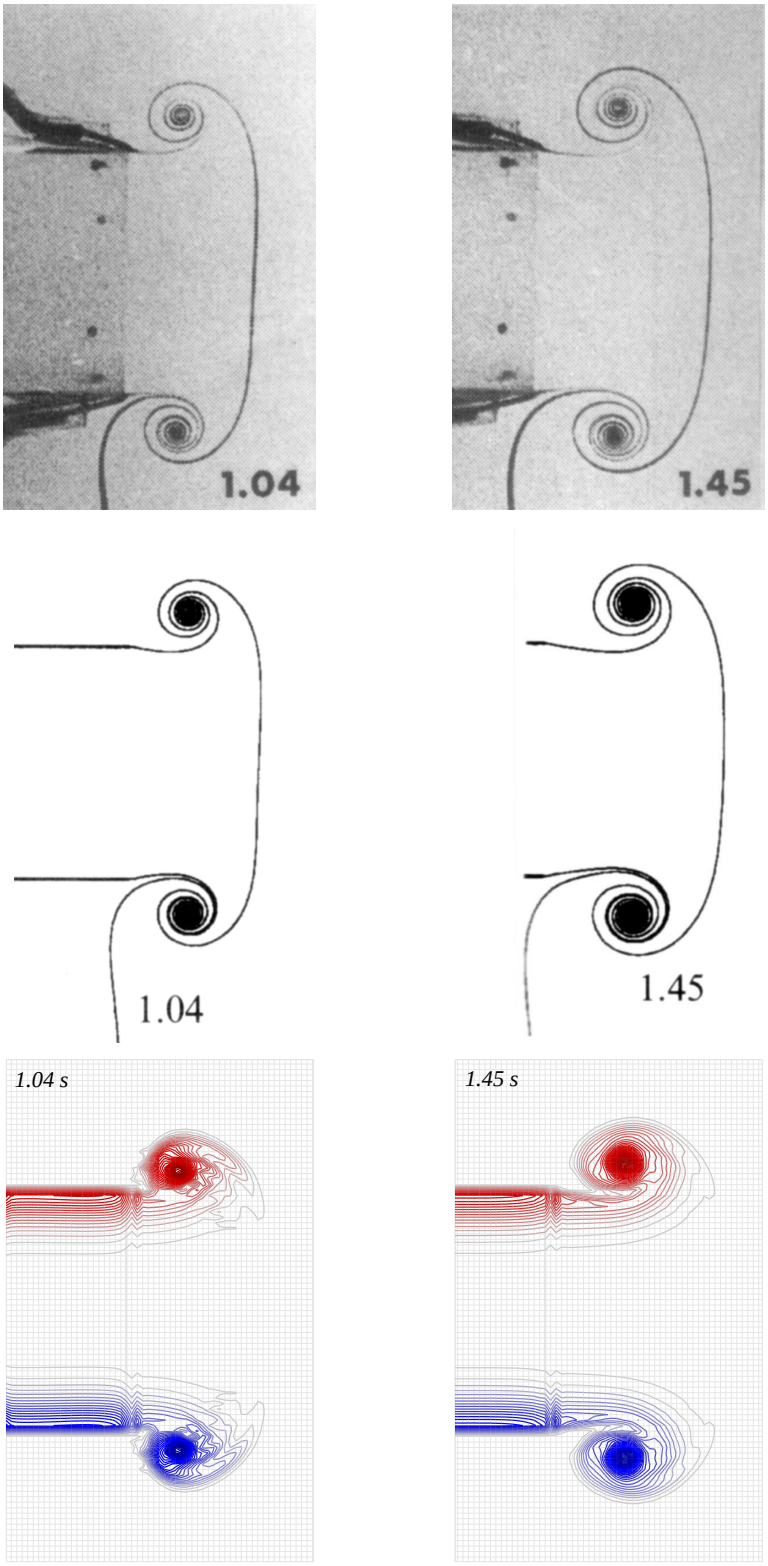


Figure 6.15: Visual comparison between Didden's experiment *(top)*, Nitsche & Krasny's numerical model ($\delta = 0.2$) *(middle)* and current study *(bottom)* for vortex ring generation problem.

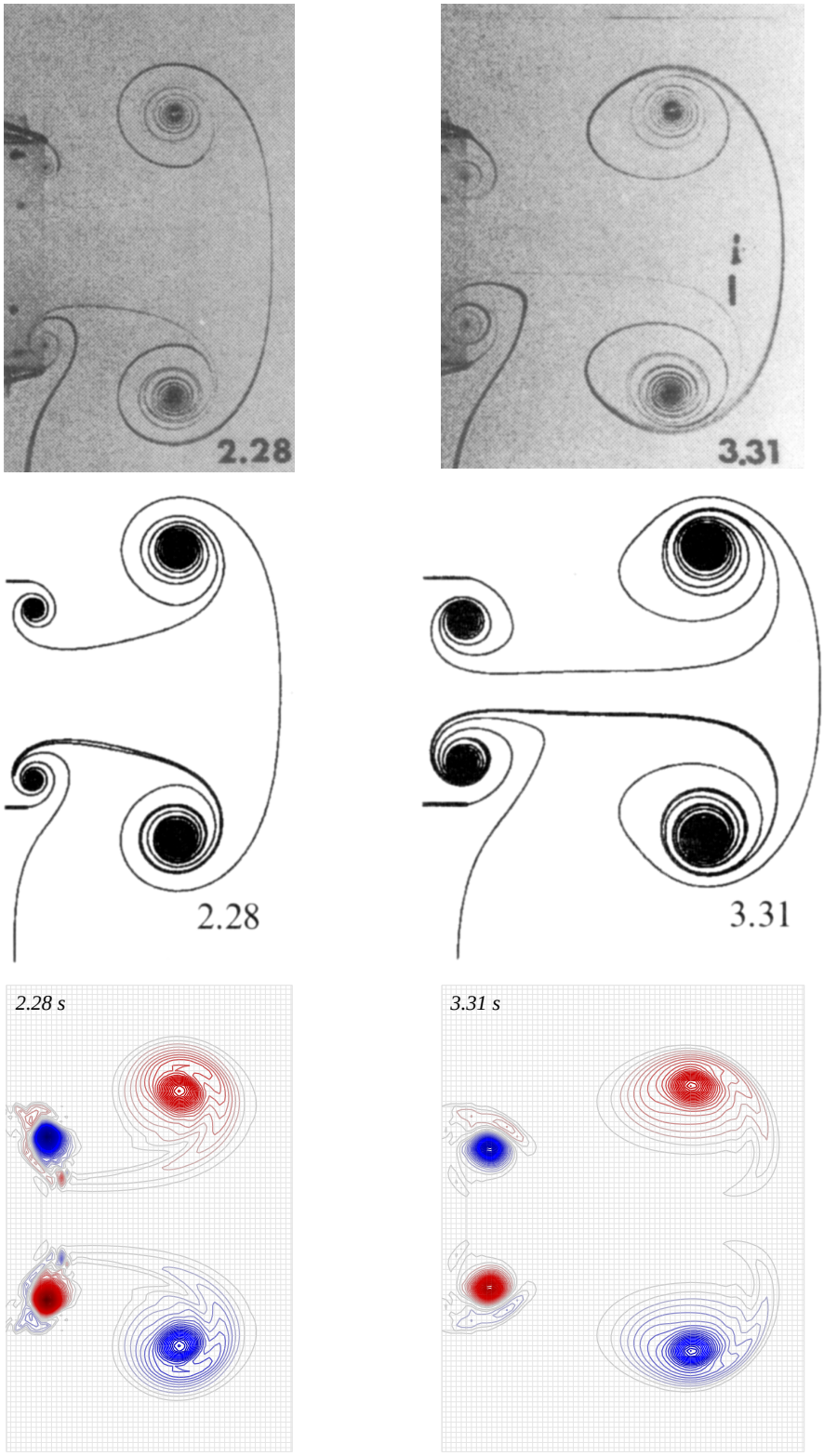


Figure 6.16: Fig. 6.15 *cont.*

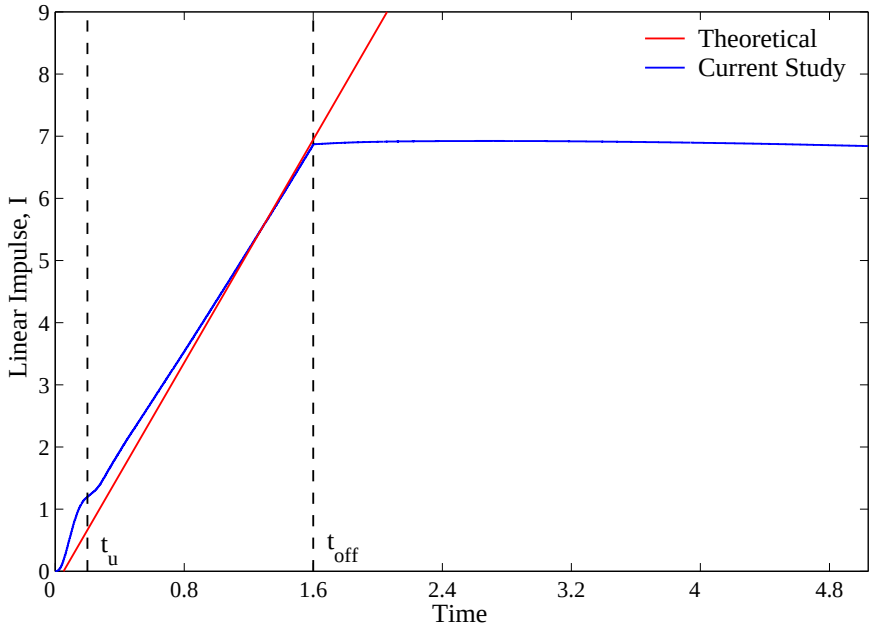


FIGURE 6.17: Measured and theoretical linear impulse

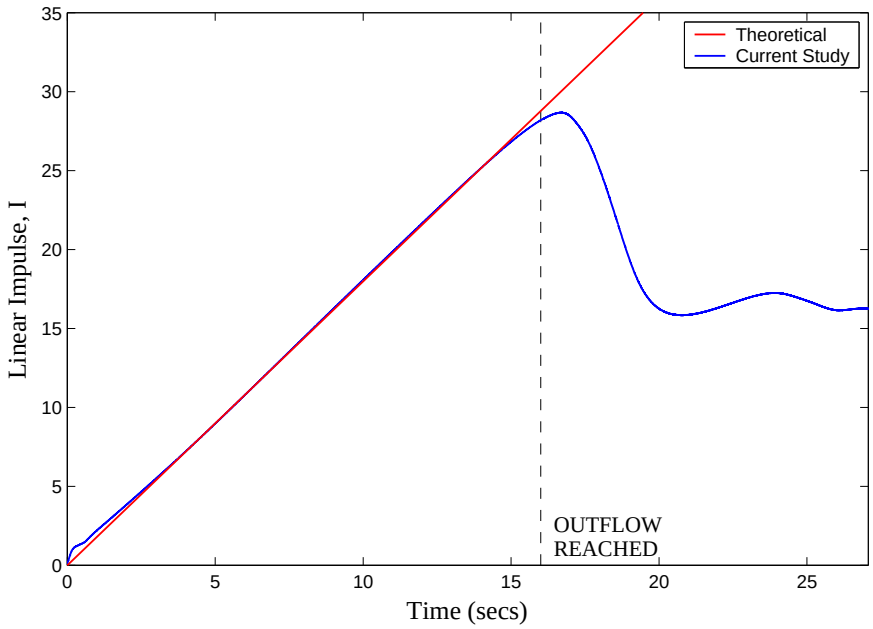


FIGURE 6.18: Measured and theoretical linear impulse

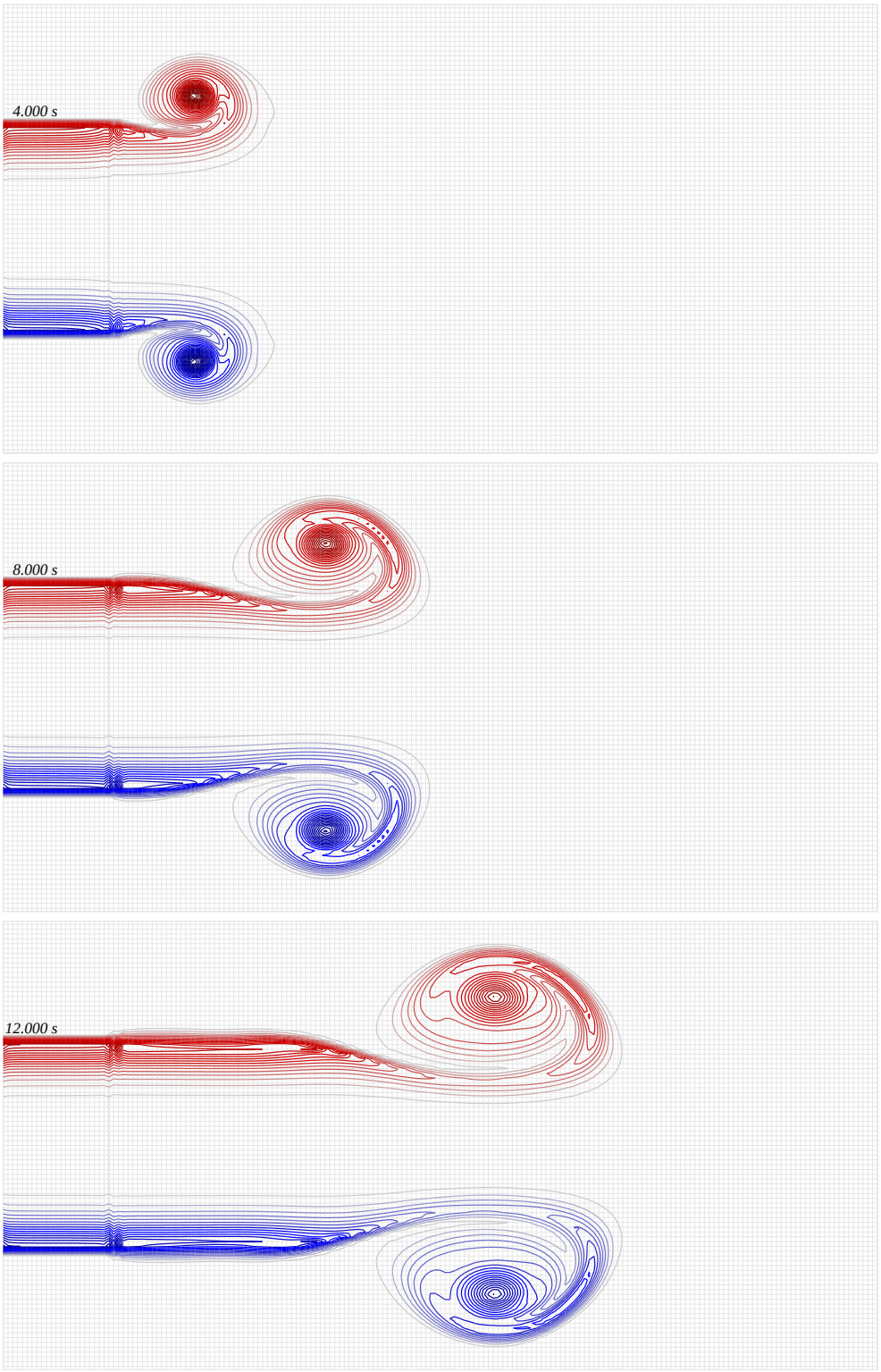


Figure 6.19: Azimuthal vorticity contours

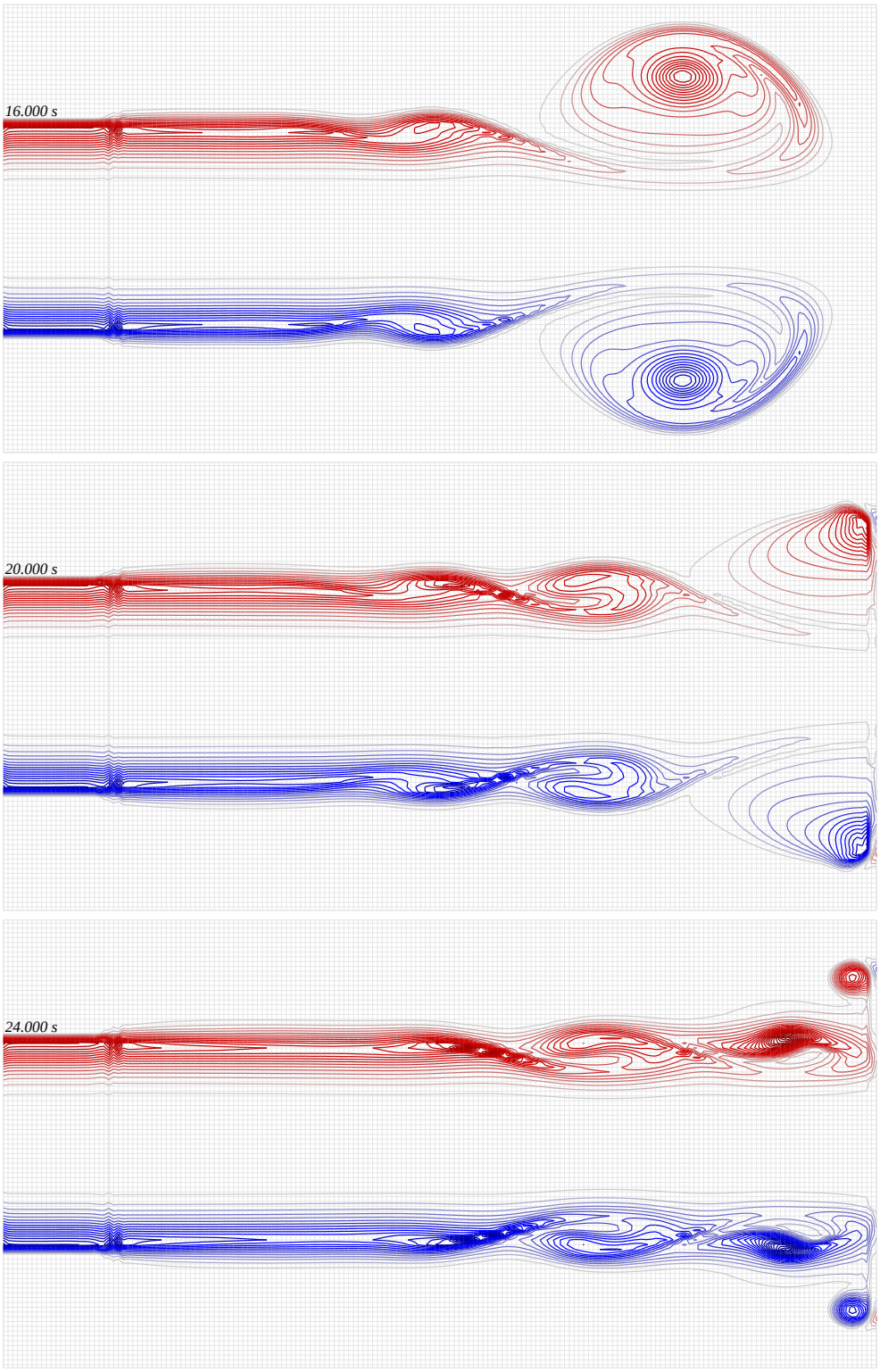


Figure 6.20: Figure 6.19 *cont.*

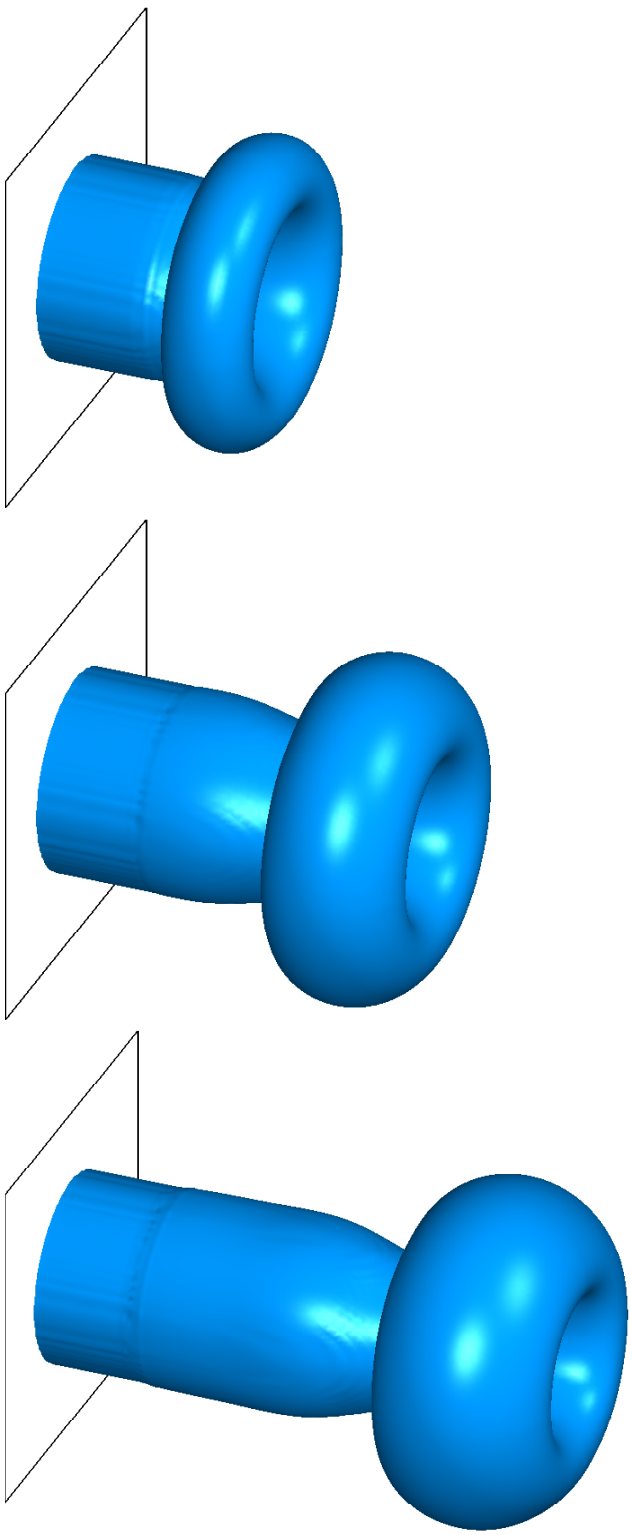


Figure 6.21: Vorticity magnitude isosurfaces

6.7 Discussion & Conclusions

A computational code for solving external flows using vortex particle-in-cell techniques has been presented. It has been demonstrated to work adequately well for the obliquely colliding rings problem, displaying excellent stability with no artificial solution regularisation required.

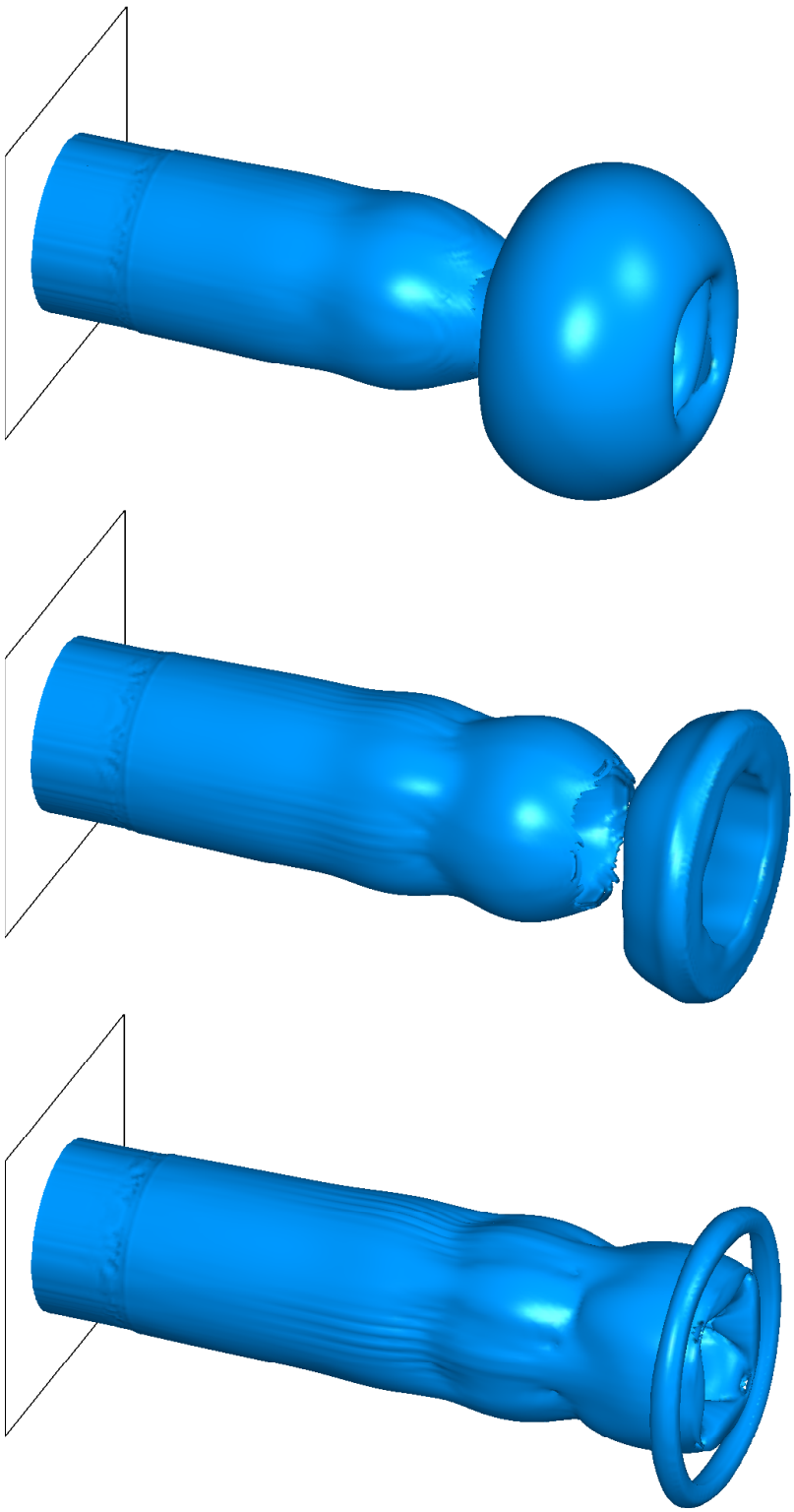


Figure 6.22: Figure 6.21 *cont.*

Appendix A

Numerical Methods

A.1 Interpolating B-Splines

– Applications of interpolation functions

– Conservation of invariants

– Necessity of regridding

– Frequency of regridding

– Derivation from convolution of top-hat

– Errors induced by regridding

– Errors induced by interpolation of data

Particle-in-cell techniques, especially in 3D, rely heavily on the ability to transfer data such as velocity and vorticity back and forth between the grid and the particle locations. This must be achieved accurately and efficiently. These transfers are most simply done by interpolation.

Also, as a consequence of the strain induced on the particle locations by the high-Re nature of the flows, it is critical to redistribute the particle locations onto a regular distribution. This can also be achieved using the interpolating B-splines outlined below.

The cubic spline M'_4, shown in Fig. A.1 is used for all such transfers. It was first developed by Monaghan for use in smoothed particle hydrodynamics [33].

$$M'_4(\mathbf{x}) = \begin{cases} 0 & \text{if } |x| > 2 \\ \frac{1}{2}(2-|x|)^2(1-|x|) & \text{if } 1 \leq |x| \leq 2 \\ 1 - \frac{5x^2}{2} + \frac{3|x|^3}{2} & \text{if } |x| \leq 1 \end{cases} \tag{A.1}$$

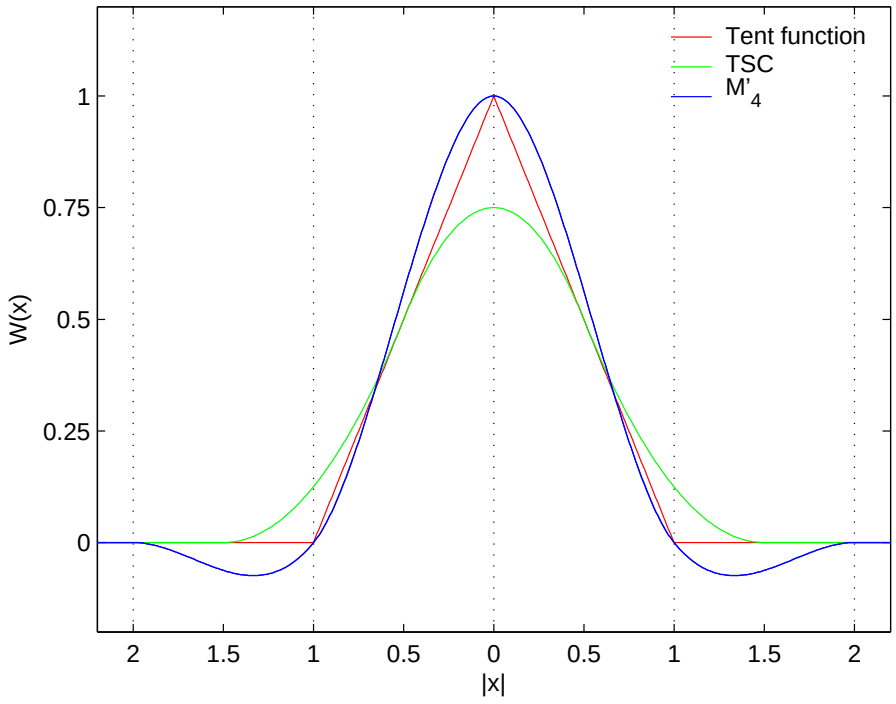


Figure A.1: B-splines of increasing order used in interpolation

When applied to 3D, all of these 1D interpolation functions become tensor products, with particle stencils of size N^d, where N is the number of particles involved in the 1D stencil, and d is the problem dimension.

This is a C^1 continuous function which spreads data to the 64 (in 3D) nearest particles. Its non-dissipative nature is shown by the fact that it conserves the angular impulse. The truncation error is thus of order h^3 where h is the distance between particles.

The other two interpolation functions shown in Fig. A.1 are given by the following formulas,

Tent function, associated with the cloud-in-cell interpolation scheme

$$M_2(\mathbf{x}) = \begin{cases} 0 & \text{if } |x| > 1 \\ 1 - |x| & \text{if not} \end{cases} \tag{A.2}$$

Triangular-shaped cloud (TSC) piecewise quadratic function

$$M_3(\mathbf{x}) = \begin{cases} 0 & \text{if } |x| > \frac{3}{2} \\ \frac{1}{2}(-|x| + \frac{3}{2})^2 & \text{if } \frac{1}{2} \leq |x| \leq \frac{3}{2} \\ \frac{1}{2}(x + \frac{3}{2})^2 - \frac{3}{2}(x + \frac{1}{2})^2 & \text{if } 0 \leq |x| \leq \frac{1}{2} \end{cases} \tag{A.3}$$

A.2 Spatial discretisation

It is necessary to calculate first order derivatives of the velocity field for stretching term calculations, and the Laplacian of the vorticity field for viscous diffusion calculations. Particularly with the stretching term, it is critical to minimise round-off error, as this error leads directly to the introduction of vorticity divergence. A 4th-order 5-point central difference, as defined below, is used throughout for first order derivatives.

Starting with the standard central difference formula,

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} - \frac{\Delta x^2}{6} f'''(x) + \cdots \tag{A.4}$$

Using a *different* step size can obtain a second approximation to the derivative. Then can algebraically combine these two expressions to eliminate the leading term of the error.

$$f'(x) = \frac{f(x+2\Delta x) - f(x-2\Delta x)}{4\Delta x} - \frac{4\Delta x^2}{6} f'''(x) + \cdots \tag{A.5}$$

$$f'(x) = \frac{f(x-2\Delta x) - 8f(x-\Delta x) + 8f(x+\Delta x) - f(x+2\Delta x)}{12\Delta x} + O(\Delta x^4) \tag{A.6}$$

The error is given by

$$\mathbb{E}(\Delta x) = \frac{\Delta x^4}{30} f^5(x) + \cdots \tag{A.7}$$

Terms with odd powers of Δx are absent, a consequence of the symmetry of the original central difference expression (A.4).

Eq. A.6 is used throughout for evaluation of first-order derivatives, both of velocity and vorticity. However, there is clearly a zone around the edge of the grid where this expression is not applicable, and non-symmetric expressions for the derivatives must be derived. The expression for a point adjacent to an edge is,

$$f'(x) = \frac{-3f(x-\Delta x) - 12f(x) + 18f(x+\Delta x) - 6f(x+2\Delta x) + f(x+3\Delta x)}{12\Delta x} + O(\Delta x^4) \tag{A.8}$$

The Laplacian operator is calculated using a 27-point compact stencil, as illustrated in Fig. A.2

A.2.1 Multidirectional correction

It is possible to perform a correction for the calculation of derivatives if the mesh is perfectly regular. This will always be the case for the xy-plane of the mesh. By

combining the standard FD stencil with a stencil rotated by 45 °, the directional dependence of the error can be greatly reduced.

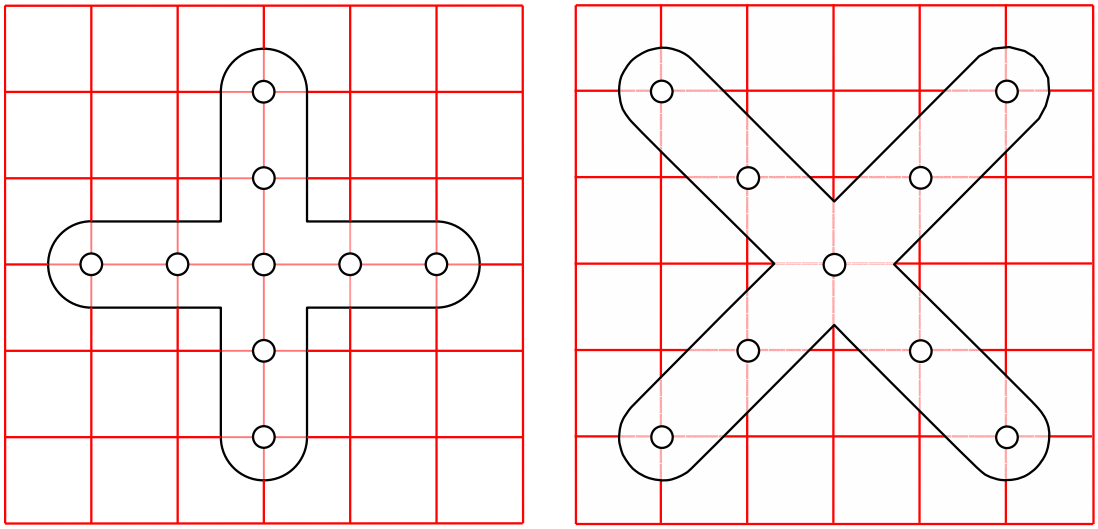


Figure A.3: Straight and rotated FD stencils

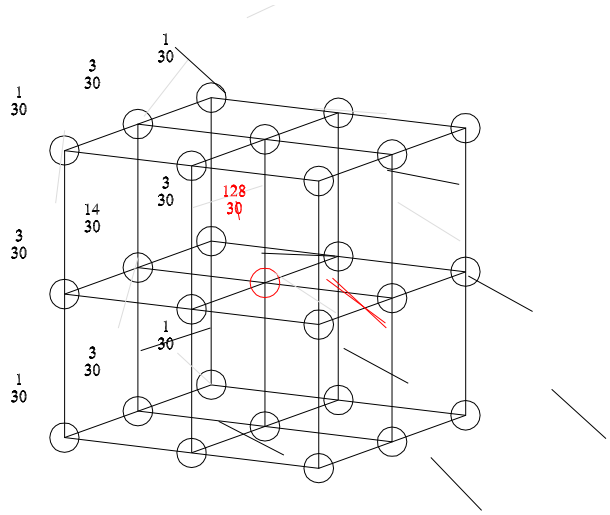


Figure A.2: Stencil of grid points for use in calculating $\nabla^2\boldsymbol{\omega}$ *(weights normalised by spacing)*

A.3 Conservation laws

The most important criterion for estimation of the accuracy of a solution is how well it conserves natural invariants of the flow

In all conservation analysis, three-dimensional unbounded flow is assumed, with zero vorticity at infinity. All of these expressions are normalised to unit density. The three linear invariants of the flow are the total vorticity, linear (or hydrodynamic) impulse and angular impulse, corresponding to the zeroth, first and second order moments of the vorticity field

$$\boldsymbol{\Omega} = \int \boldsymbol{\omega}\,\mathrm{d}\mathbf{x} = 0 \tag{A.9}$$

$$\boldsymbol{I} = \int \mathbf{u}\,\mathrm{d}\mathbf{x} = \frac{1}{2}\int \mathbf{x}\times\boldsymbol{\omega}\,\mathrm{d}\mathbf{x} \tag{A.10}$$

$$\boldsymbol{A} = \int \mathbf{x}\times\mathbf{u}\,\mathrm{d}\mathbf{x} = \frac{1}{2}\int \mathbf{x}\times(\mathbf{x}\times\boldsymbol{\omega})\,\mathrm{d}\mathbf{x} \tag{A.11}$$

These are defined *per unit density* of the flow. The $\frac{1}{2}$ in front of the integral in the expression for impulse disappears in 2D. One other quadratic invariant of importance is the total kinetic energy

$$\boldsymbol{E} = \int (\mathbf{u}\cdot\mathbf{u})\mathrm{d}\mathbf{x} \tag{A.12}$$

For inviscid flow, all of these invariants should be conserved. The addition of viscosity, however, will clearly lead to a reduction in kinetic energy with time. The rate of this reduction is derived in [38] as follows,

Taking the dot product of $\mathbf{u}$ s the momentum equation and integrating over an unbounded volume, the rate of change of $\boldsymbol{E}$ is

$$\frac{\mathrm{d}\boldsymbol{E}}{\mathrm{d}\,t} = -2\nu\int \mathbb{S}_{ij}\mathbb{S}_{ij}\,\mathrm{d}\mathbf{x} = -\int \Phi\,\mathrm{d}\mathbf{x} \tag{A.13}$$

where $\mathbb{S}_{ij}$ was defined in § 3.3 as the rate-of-strain tensor, and Φ is the *dissipation function.* It can be shown that

$$\Phi = \nu\boldsymbol{\omega}\cdot\boldsymbol{\omega} - \nu\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u}) \tag{A.14}$$

For unbounded flows, this just leaves us with

$$\frac{\mathrm{d}\boldsymbol{E}}{\mathrm{d}\,t} = -\nu\mathcal{E}_T \tag{A.15}$$

where $\mathcal{E}_T$ is the total enstrophy, defined as

$$\mathcal{E}_T = \int \boldsymbol{\omega}\cdot\boldsymbol{\omega}\,\mathrm{d}\mathbf{x} \tag{A.16}$$

The discrete form of this expression is

$$\mathcal{E}_p = \sum_p \mathrm{vol}_p|\boldsymbol{\omega}_p|^2 \tag{A.17}$$

So, the rate of change of kinetic energy is given by Eq. A.15 and is always negative. Enstrophy itself is also not conserved in three-dimensional flows, due to the stretching term. Its rate of change is determined in a similar way to that of the kinetic energy. The dot product of $\boldsymbol{\omega}$ with the vorticity equation Eq. 3.21 is taken, with the final result

$$\frac{\mathrm{d}\mathcal{E}}{\mathrm{d}\,t} = 2\int \boldsymbol{\omega}_i\boldsymbol{\omega}_j\mathbb{S}_{ij}\,\mathrm{d}\mathbf{x} - 2\nu\int \frac{\partial\boldsymbol{\omega}_i}{\partial x_j}\frac{\partial\boldsymbol{\omega}_i}{\partial x_j}\,\mathrm{d}\mathbf{x} \tag{A.18}$$

Comparison between the left and right hand sides in the expression for the rate of change of kinetic energy is especially useful as it measures how accurately the viscous term is being modelled. Any discrepancy between the two sides will give an estimate of the numerical dissipation.

Bibliography

[1] T.T. Lim. A note on the leapfrogging between two coaxial vortex rings at low reynolds numbers. *Physics of Fluids*, 9(1):239–241, 1997.

[2] T.T. Lim. An experimental study of a vortex ring interacting with an inclined wall. *Experiments in Fluids*, 7:453–463, 1989.

[3] G.S. Winckelmans and A. Leonard. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *Journal of Computational Physics*, 109:247–273, 1993.

[4] N. Didden. On the formation of vortex rings: rolling-up and production of circulation. *Z. Agnew. Math. Phys.*, 30:101, 1979.

[5] L. Rosenhead. The formation of vortices from a surface of discontinuity. *Proceedings of the Royal Society – Series A*, 134:170, 1931.

[6] Niklas Andersson, Lars-Erik Eriksson, and Lars Davidson. A study of Mach 0.75 jets and their radiated sound using large-eddy simulation. In *Proceedings of the Tenth AIAA/CEAS Aeroacoustics Conference, May 10-12 2004, Manchester, United Kingdom*, 2004.

[7] A.J. Chorin. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics*, 57:380, 1973.

[8] P. Koumoutsakos and A. Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics*, 296:1, 1995.

[9] J. Barnes and P. Hut. A hierarchical O(NlogN) force-calculation algorithm. *Nature*, 324(6096):446–449, 1986.

[10] L. Greengard and V. Rohklin. A fast algorithm for particle simulation. *Journal of Computational Physics*, 73:325–348, 1987.

[11] Keith Lindsay and Robert Krasny. A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *Journal of Computational Physics*, 172:879–907, 2001.

[12] J.P. Christiansen. Vortex methods for flow simulation. *Journal of Computational Physics*, 13:363, 1973.

[13] Charles K. Birdsall and Dieter Fuss. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation. *Journal of Computational Physics*, 3:494, 1969.

[14] Conor Sweeney. *A new technique for vortex method in computational fluid dynamics*. PhD thesis, Trinity College, Dublin, 2001.

[15] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. *Math. Comp.*, 53:485–526, 1989.

[16] A. Leonard. Numerical simulation of interacting, three-dimensional vortex filaments. In Heidelberg: Springer-Verlag, editor, *Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics*, pages 245–250, 1975.

[17] A. Leonard. Vortex methods for flow simulation. *Journal of Computational Physics*, 37:289–335, 1980.

[18] A. Leonard. Computing three-dimensional incompressible flows with vortex elements. *Annual Review of Fluid Mechanics*, 17:523–559, 1985.

[19] C. Rehbach. Calcul d'ecoulements autour d'ailes sans epaisseur avec nappes tourbillonnaires evolutives. *La Recherche Aerospatiale*, 2:53–61, 1973. English translation NASA TT-F 1583.

[20] C. Rehbach. Numerical calculation of three-dimensional unsteady flows with vortex sheets. In *Proceedings of 16th AIAA Aero. Sci. Meeting, Huntsville, Alabama*, 1978.

[21] B. Couet, O. Buneman, and A. Leonard. Simulation of Three dimensional incompressible flows with a Vortex-in-Cell method. *Journal of Computational Physics*, 39:305–328, 1981.

[22] Stephen H. Brecht and John R. Ferrante. Vortex-in-cell simulation of buoyant bubbles in three dimensions. *Physics of Fluids A*, 1(7):1166–1191, 1989.

[23] Stephen H. Brecht and John R. Ferrante. Vortex-in-cell calculations in 3 dimensions. *Computer Physics Communications*, 58(1):25–54, 1990.

[24] G-H. Cottet and P. Poncet. Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods. *Journal of Computational Physics*, 193(1):136–158, January 2004.

[25] P.G. Saffman. *Vortex Dynamics*. Cambridge, 1992.

[26] P. Swarztrauber and R. Sweet. Efficient FORTRAN subprograms for the solution of Elliptical Differential Equations. NCAR Technical Note-TN/IA-109, July 1975.

[27] P. Orlandi. *Fluid Flow Phenomena - A Numerical Toolkit*. Kluwer Academic Publishers, 2000.

[28] G-H. Cottet and P. Koumoutsakos. *Vortex Methods - Theory and Practice*. Cambridge University Press, 2000.

[29] G.S. Winckelmans. *Topics in vortex methods for the computation of three- and two-dimensional incompressible unsteady flows.* PhD thesis, California Institute of Technology, 1989.

[30] J.H. Williamson. Low-storage Runge-Kutta schemes. *Journal of Computational Physics*, 35:48–56, 1980.

[31] G. Pedrezetti. Insight into singular vortex flows. *Fluid Dynamics Research*, 3:CHECK!!, 1981.

[32] M.L. Ould-Salihi, G-H. Cottet, and M. El Hamraoui. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM Journal of Scientific Computing*, 22(5):1655–1674, 2000.

[33] J.J. Monaghan. Extrapolating B-splines for interpolation. *Journal of Computational Physics*, 60:253–262, 1985.

[34] M. Nitsche and R. Krasny. A numerical study of vortex ring formation at the edge of a circular tube. *Journal of Fluid Mechanics*, 276:139–161, 1994.

[35] O.M. Knio and A.F. Ghoniem. Numerical study of a three dimensional vortex method. *Journal of Computational Physics*, 86:75–106, 1990.

[36] K. Shariff, R. Verzicco, and P. Orlandi. A numerical study of three-dimensional vortex ring instabilities: viscous corrections and early nonlinear stage. *Journal of Fluid Mechanics*, 279:351–375, 1994.

[37] E. Rivoalen and S. Huberson. Numerical simulation of axisymmetric viscous flows by means of a particle method. *Journal of Computational Physics*, 152:1–31, 1999.

[38] G.K. Batchelor. *An Introduction to fluid dynamics.* Cambridge University Press, 1967.

Index