Diss. ETH No. 22233

3D simulations of vortex dynamics and biolocomotion

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

WILLEM MARINUS VAN REES

M.Sc., Delft University of Technology
born on October 29th, 1983
citizen of the Netherlands

accepted on the recommendation of

Prof. P. Koumoutsakos, examiner
Prof. J.O. Dabiri, co-examiner
Prof. A. Leonard, co-examiner

2014

Abstract

Vortical flows are crucial in applications ranging from aircraft takeoff to fish swimming. Numerical flow simulations can increase our understanding of such phenomena in order to improve the performance of engineering devices and characterize natural processes. Yet simulations of unsteady 3D flows on modern computer architectures face several challenges. The accuracy and validity of the numerical methods need to be established before physical processes can be predicted with confidence. Efficient simulations demand high-performance software that can adapt computational elements according to the spatiotemporal scales in the flow. Physical relevance is obtained only when the resulting flow data is translated into knowledge of the dominating structures and interactions through analysis and post-processing. In this work, we contribute to each of these issues.

We use the remeshed vortex method for the simulation of selected 3D flow cases. We validate the method with vortex dynamics computations at various Reynolds numbers, and show that similar accuracy to the established pseudo-spectral method can be achieved. The solver is then applied to vortex reconnection, a mechanism for energy decay in turbulent flows, at an unprecedented Reynolds number of 10^4. An analysis of the results uncovers novel flow features such as axial flow generation and secondary reconnection instabilities.

The solver is extended with fluid-structure interaction capabilities and coupled to an optimization algorithm to create a versatile reverse-engineering framework. This approach is then used to identify the optimally fast and efficient morphologies and kinematics of 3D undulatory swimmers. The optimal solutions outperform a natural reference swimmer by reaching almost three times higher speeds and over four times higher efficiencies. We investigate how the optimal swimmers manipulate the fluid to accomplish these feats.

Lastly, we develop a 3D multiresolution remeshed vortex method to improve the computational efficiency of fluid-structure interaction simulations. It is designed for high performance on modern compact compute platforms and employs a hybrid CPU/GPU velocity solver. The software enables the first ever computation multiple 3D self-propelled swimmers and their hydrodynamic interactions, achieved using only four compute nodes.

Zusammenfassung

Wirbelströmungen spielen eine wichtige Rolle in so verschiedenen Prozessen wie Flugzeugstarts oder schwimmenden Fischen. Strömungssimulationen können unser Wissen und Verständnis solcher Strömungen vertiefen, und damit natürliche Prozesse erklären oder die Leistung technischer Anwendungen verbessern.

Simulationen von instationären 3D Strömungen auf modernen Computern bergen mehrere Herausforderungen. So muss die Genauigkeit der numerischen Methoden nachgewiesen werden, bevor physikalische Prozesse zuverlässig vorausgesagt werden können. Effiziente Simulationen benötigen ausserdem Hochleistungssoftware und die Möglichkeit, Rechenelemente den raumzeitlichen Skalen der Strömung anzupassen. Physikalisch sinnvolle Aussagen sind am Ende nur möglich, wenn ein Verständnis durch sorgfältige Analyse und Nachbearbeitung der resultierenden Strömungsdaten hergestellt wird.

In der vorliegenden Arbeit verwenden wir die 'Remeshed-Vortex-Methode' um ausgewählte 3D Strömungen zu simulieren. Wir zeigen die Genauigkeit der Methode für Wirbelströmungen im Vergleich zu einer gut validierten 'Pseudo-Spektral-Methode'. Dann simulieren wir 'Vortex Reconnection' – die Verbindung zweier Wirbelrohre – und untersuchen ihre Rolle für den Energiezerfall in turbulenten Strömungen.

Die numerische Methode wurde mit Fluid-Struktur-Interaktion Techniken erweitert und mit einem evolutionären Optimierungsalgorithmus kombiniert. Mit dieser Reverse-Engineering Rahmenstruktur finden wir die schnellsten und effizientesten Formen und Bewegungen für 3D Schwimmer. Unsere optimalen Schwimmer erreichen fast dreimal höhere Geschwindigkeit oder mehr als viermal höhere Effizienz als gewisse existierende Fische. Wir analysieren die Wirbel und Fliessstrukturen um zu erklären, wie diese optimalen Schwimmer die Strömung für ihre Leistung nutzen.

Schliesslich entwickelten wir einen 3D-Multi-Resolutions-Solver für die 'Remeshed-Vortex-Methode', um solche Fluid-Struktur-Interaktions-Probleme effizienter simulieren zu können. Die Software wurde so entworfen, dass sie auf modernen Computern inklusive Grafikkarten eine hohe Leistung erreicht. Mit diesem Solver konnten wir erstmals die hydrodynamischen Interaktionen zwischen mehreren 3D Schwimmern simulieren.

Acknowledgments

I wish to thank my supervisor, Petros Koumoutsakos, for giving me as a fellow naval architect the opportunity to join his computational science and engineering group. His guidance during these years has instilled in me the importance of working with pride, care and confidence. He continuously facilitated interactions with renowned visitors in Zurich, and allowed me to travel and meet with distinguished scientists around the world. These experiences have greatly contributed to my development, and I'm very grateful for the opportunities he has provided me.

I thank Tony Leonard, who has overseen and supported my work all the way from co-authoring my first paper to serving in the committee of my thesis defense. It has been a privilege to interact with such an open-minded and kind pioneer of fluid dynamics. My gratitude also goes to John Dabiri, for the several interactions we had during my visits at Caltech and for serving in my doctoral committee.

I am indebted to Dale Pullin and Fazle Hussain for their collaborations on the vortex reconnection studies, and special gratitude to Dale and Joseph Shepherd for hosting my visits to Caltech. I thank Georges-Henri Cottet and Jens Walther for the many open and instructive discussions on particle methods, remeshing and boundary conditions.

I wish to thank all past and current members of the CSE-lab, who have provided me with incredible inspiration, knowledge and joy during and after work. Specifically, Philippe and Michael set the scientific examples to strive for. Diego and Panos H. have been vital references on computer science and uncompromising developments, and Diego has established the basis for the multiresolution results and the nicest visualizations in this thesis. I collaborated especially closely with Mattia, whose sharp-wittedness and ingenuity turned our shared projects into key learning experiences for me. Panos A. has been an infinite source of more and mostly less appropriate remarks, but above all has been a good friend during our shared years in the group. Finally Babak, with whom I have interacted many times each day, both in and out of the lab. I have tremendously enjoyed our rants, jokes and discussions on anything ranging from good science to bad wine and vice versa.

Some of my most precious moments in Zurich were my early morning rowing practices out on the lake, and I wish to thank the Gemischter 8er for their enthusiasm. Particularly Florian, our evening concert visits and post-rowing coffee talks have been valuable intermezzos for me.

My bike trips throughout Switzerland would not have been the same without the countless roads and mountain passes I've ridden next to, in front of, and behind Christophe. You have surrounded me with lots of humor, a critical external perspective, and, together with Nuria, an always welcoming and friendly environment for me to enjoy your company.

Jacqueline, you were the very first person I met in Zurich, and the most special of all. Thank you for your enduring support, patience and love, they have brightened my life in immeasurable ways.

The support of my family has been invaluable. The endless support and care provided by my parents keeps enabling me to grow and try to live up to their example. Lastly, Balt, Hans, Cees and Sjoerd, I wish I could spend more time with all of you. It is a great inspiration to see you guys combine incredible dedication with contagious silliness to approach life in the best way possible.

Contents

CHAPTER 1

Introduction

1.1 Vortical flows

Vortical flows can capture our fascination as we watch fragile smoke twirling from an extinguished candle, or gape at the relentless swirling of strong river currents under a bridge. Between such fleeting observations it is easy to forget that these flows are critical to many impressive feats of nature and engineering. Biological creatures such as birds and fishes can interact with the fluid around them and achieve an agility that is still unrivaled by man-made robotic devices. Try to catch a small fish bare-handedly, and see how quickly it can maneuver and accelerate to escape. Similarly, the ability to understand and exploit fluid dynamics is crucial to a successful design of modern engineering machines such as aircraft, turbomachinery and wind turbines.

Despite many decades of intensive study, vortical flows continue to pose great scientific challenges. The fundamental physical mechanisms that govern turbulent flows are still not fully understood. And an answer to the question whether the fluid dynamics equations can develop into a singular solution will be rewarded by one million dollars – one of the currently six open Millennium Prize problems.

The incompressible flows in these examples are described by the Navier-Stokes equations, a set of partial differential equations representing the spatiotemporal evolution of fluid momentum. Using suitable numerical techniques and computational resources these equations can produce accurate predictions for the evolution of a fluid flow. Through continuing developments in hardware, software, and numerical methods, modern computational tools possess the capabilities to accurately simulate a subset of relevant engineering and scientific problems. Such a computational approach can enable a level of flexibility and detail unattainable with experiments.

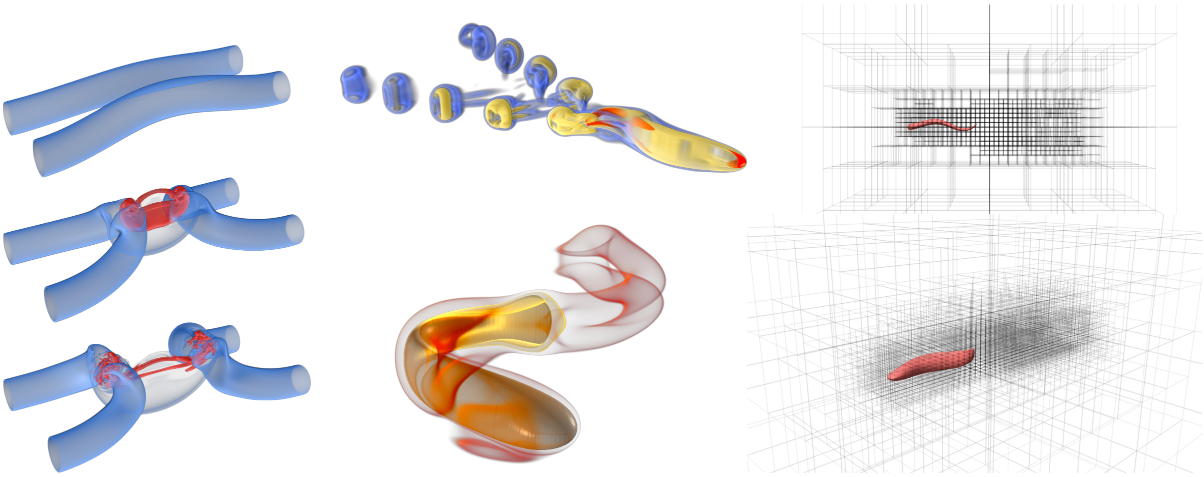


Figure 1.1: An overview of the topics considered in this thesis: vortex reconnection (left), biolocomotion (middle) and multiresolution flow simulations (right).

Vortex methods were among the first techniques used for simulations of 3D vortical flows and have since been adapted to ensure convergence and improve computational efficiency. One of the results is the remeshed Vortex Method (rVM), which combines the advantages of using Lagrangian particles for convection with the efficiency enabled by Eulerian regular grids.

In this thesis we study and employ the rVM for vortical flow simulations, using two complementary perspectives. By advancing the state of the art in 3D simulations of vortical flows we aim to contribute to a toolset that can be used for a wide variety of relevant applications, beyond those considered in this thesis. We then apply these tools to accomplish unprecedented 3D simulations of selected topics in vortex dynamics and biolocomotion. Through detailed analysis and interpretation of the results we advance the physical understanding in these fields.

This thesis consists of three parts (figure 1.1). First, we consider the simulation of vortex dynamics in the absence of solid obstacles. We study vortex reconnection, one of the archetypal problems in fluid dynamics with relevance to the fundamentals of turbulence and the question of a finite-time singularity in the Navier-Stokes equations. Second, we deal with the interaction between a fluid and immersed solid obstacles. We consider the numerical requirements of fluid-solid interaction (FSI) simulations and their coupling with an evolutionary optimization algorithm. We study the optimality of 3D self-propelled swimmers, and the physical mechanisms behind undulatory aquatic locomotion. Third, we consider how multiresolution techniques can exploit the large separation in spatiotemporal scales of such FSI problems. We provide an efficient implementation of a multiresolution

3D FSI solver on modern multi/many-core computational hardware, and present its validation and performance.

1.2 State of the art and challenges

This thesis aims to advance our current numerical and physical knowledge of 3D simulations of vortex reconnection, FSI and optimization of self-propelled swimmers, and multiresolution simulations. In this section we will discuss the state of the art and challenges in each of these topics, distinguishing between the numerical and the physical perspectives on the topics.

1.2.1 High Reynolds number vortex dynamics

The Reynolds number represents the ratio between inertial and viscous effects in a flow. As the Reynolds number increases, computational requirements to capture the spatiotemporal range from the largest inertial flow structures down to the smallest dissipative scales increase correspondingly. In this thesis we consider specifically the case of vortex reconnection, a fundamental fluid dynamics problem that places rigorous demands on the numerical method. To rely on the results of numerical simulations for such challenging flows requires extensive verification of the method.

Numerics The rVM has been proven accurate for three-dimensional flows with Reynolds numbers up to 3500 in a comparison study with a pseudo-spectral method (Cottet, Michaux, et al. 2002). For higher Reynolds number flows and in particular for the demanding case of vortex reconnection, which is characterized by strong gradients and singular-like behavior, detailed results on performance and accuracy of the rVM have not been shown. Since the rVM relies on an interpolation kernel to combine the Lagrangian particles with the Eulerian grid, it is of particular interest to compare different kernel functions. In the context of flow simulations, such a comparison has only been presented for 2D inviscid flow problems (Koumoutsakos 1997).

Simulating the broad range of spatiotemporal scales in high Reynolds number 3D vortical flows places strong demands on memory and computing power. Practically this translates into the use of supercomputing systems that offer large-scale distributed-memory compute architectures. On such systems both the simulation data and computational work can be distributed between the compute nodes. Such parallelism comes at the cost of an inter-node communication overhead that reduces the efficiency of the code. The large problem size also complicates the post-processing and analysis, since the full data files can only be analyzed and visualized using distributed-memory software and hardware. These matters have to be overcome to provide answers to timely questions in high Reynolds number vortex dynamics.

Physics The repeated occurrence of vortex reconnections has been postulated as a driving mechanism for turbulence and aerodynamic noise generation (Hussain 1983) and the destruction of aircraft wakes (Crow 1970; Scorer and Davenport 1970). In the last decades, studies of vortex reconnection have been central to the question of singularity formation of both Euler and Navier-Stokes equations (Pumir and Siggia 1987; Kerr 1993; Hou and Li 2007; Hou and Li 2008; Hou and Lei 2009). Several previous numerical studies, based almost exclusively on spectral methods, have expanded our knowledge for circulation-based Reynolds numbers up to 3500, at least up to the first reconnection event (Melander and Hussain 1988; Shelley, Meiron, and Orszag 1993). Results of vortex reconnection at Reynolds numbers up to 9000 were presented during this thesis by Hussain and Duraisamy (2011), although without a thorough analysis of the dominating physical mechanisms. Results for the long-time evolution of reconnected structures, particularly relevant to the hypothesis of cascading vortex reconnections, have not been reported before.

A more general challenge associated with studying high Reynolds number 3D vortex dynamics is to identify the mechanisms driving the flow dynamics. Simply observing the vorticity and velocity fields will not provide this insight as the multitude of small scales clouds the dominating processes. Instead, analyzing such flows relies on combining the insight from several metrics such as energy spectra, temporal energy and enstrophy evolution, and the visualization of derived quantities specifically tailored to each flow case.

1.2.2 FSI and optimization of self-propelled swimmers

Two-way FSI combines the response of a fluid to the presence of a body with the temporal evolution of the body due to the fluid forces on its surface. An FSI solver needs to integrate these effects accurately and sufficiently flexible when considering its coupling with optimization techniques. Such a coupling will provide a framework to reverse-engineer optimal swimmers without strong *a priori* constraints and assumptions, which can be used to advance our understanding of biological and robotic aquatic locomotion.

Numerics Solid obstacles have traditionally been incorporated in vortex methods using body-fitted grids or panel methods (Koumoutsakos and Leonard 1995; Ploumhans and

Winckelmans 2000). These approaches treat the boundary conditions accurately but are involved and inflexible when dealing with arbitrarily shaped obstacles. An alternative approach was proposed in Coquerelle and Cottet (2008) based on Cartesian-grid volume penalization methods (Arquis and Caltagirone 1984; Angot, Bruneau, and Fabrie 1999). Here the obstacle is implicitly represented on a regular grid and no-slip boundary conditions are imposed through a forcing term in the Navier-Stokes equations. This method additionally allows the use of a projection method (Patankar and Sharma 2005) to enable two-way fluid-solid coupling. With this approach the transfer of momentum between fluid and body is directly evaluated, without the need to compute surface forces on the body. This technique was used in Coquerelle and Cottet (2008), although not for deforming obstacles such as swimmers.

Several simulations of self-propelled swimmers that do not rely on vortex methods have been presented prior to this thesis, although with the majority of works limited to 2D. Full 3D results of single deforming self-propelled swimmers based on immersed boundary/interface techniques have been presented in Gilmanov and Sotiropoulos (2005), Hieber and Koumoutsakos (2008), Mittal et al. (2008), Shirgaonkar, MacIver, and Patankar (2009), Borazjani, Sotiropoulos, et al. (2012), and Bhalla et al. (2013). Simulations of multiple three-dimensional swimmers require methods that provide support for multiple obstacles without compromising their efficiency. To the best of our knowledge, such simulations have not been presented prior to this thesis.

We note that the numerical challenges of developing an FSI solver for self-propelled swimmers are not restricted to this specific application. In fact, modern fluid dynamics engineering problems are dominated by FSI phenomena, ranging from cardiovascular fluid dynamics (Peskin 1982) to wind turbine farms (Whittlesey, Liska, and Dabiri 2010). These applications share our challenges to accurately and efficiently simulate the interaction between complex, possibly deforming solid obstacles and a fluid flow.

Optimization The Covariance Matrix Adaptation–Evolutionary Strategy (CMA-ES) (Hansen and Ostermeier 2001; Hansen, S. D. Müller, and Koumoutsakos 2003) has proven to be a robust and suitable optimization algorithm and has successfully been coupled with both low-order (Tokic and Yue 2012) and fully viscous numerical flow solvers (Kern, Hansen, and Koumoutsakos 2006; Chatelain, Gazzola, et al. 2011; Gazzola, Vasilyev, and Koumoutsakos 2011).

Physics The relation between form and function in the hydrodynamics of aquatic biolocomotion has primarily been studied from the perspective of natural and bio-inspired robotic swimmers. These studies range from gathering and analyzing high-level metrics of several species (Taylor, Nudds, and Thomas 2003; Rohr and Fish 2004; M. S. Triantafyllou, Hover, et al. 2005), to detailed hydrodynamic studies of individual swimmers (Tytell 2004a; U. K. Müller, van den Boogaart, and van Leeuwen 2008; Borazjani and Sotiropoulos 2009) and their isolated traits (Lauder 2000; Wilga and Lauder 2004; Dong et al. 2010). The restriction to naturally occurring morphologies and kinematics, however, does not necessarily provide us with a complete picture. Natural swimmers have evolved with respect to Darwinian criteria such as survival and procreation, and under constraints specified by their physiological systems and biological materials. Finding solutions that maximize hydrodynamic performance metrics such as speed or efficiency, irrespective of any such biological constraints, puts natural swimmers' performances into context. Features found in optimal solutions can explain their functionality, and can be compared to natural swimmers. Finally, the knowledge of optimal solutions can steer the design of robotic swimmers towards engineering solutions that outperform natural swimmers.

Along this line, Tokic and Yue (2012), Eloy (2013), and Kohannim and Iwasaki (2014) recently presented swimmer optimizations combining models for the energy expenditure of fish with low-order potential flow models. Such simplified flow models sacrifice accuracy and detailed insight of the flow hydrodynamics for faster solution times and broader optimization capabilities. Optimizations for fully viscous swimmers are limited to those in Kern, Hansen, and Koumoutsakos (2006), where the optimally fast and efficient motions for eel-like morphologies were found. Optimizations based on 3D viscous simulations that involve the swimmer's morphology have not been presented prior to this thesis.

1.2.3 Multiresolution 3D flow simulations

Typical FSI problems are characterized by a strong disparity in spatiotemporal scales, ranging from small scales in the boundary-layer to larger scales in the wake and to quiescent flow upstream of the body. These scales are neither spatially nor temporally static, and an efficient simulation of such flows requires dynamic adaptation of the computational elements accordingly. Such an approach can reduce both the number of computational operations and the memory footprint of the solver.

A widely used adaptive-grid technique in fluid dynamics is Adaptive Mesh Refinement (AMR, Berger and Oliger 1984; Miniati and Colella 2007). AMR supports unstructured grids and has recently been employed to simulate 3D self-propelled swimmers (Bhalla et al. 2013). On the other hand, wavelet-based multiresolution techniques for fluid mechanics applications (Schneider and Vasilyev 2010) are used for their high compression rates, capability of performing *a priori* error predictions and efficiency by means of the Fast

Wavelet Transform. Wavelet-based multiresolution techniques have first been coupled with particle methods in Bergdorf and Koumoutsakos (2006). Further development on this topic has resulted in a 2D wavelet-based adaptive rVM, based on the MRAG library developed in the CSE-Lab (Rossinelli 2011; Gazzola, Mimeau, et al. 2012; Rossinelli, Hejazialhosseini, van Rees, et al. 2015). This solver includes fluid-solid interaction and time-adaptivity by means of local time stepping schemes (Hejazialhosseini, Rossinelli, Bergdorf, et al. 2010). A prototype software providing the extension of this framework to 3D was presented in Rossinelli (2011).

The main challenge for both 2D and 3D multiresolution simulation software is to effectively implement the governing algorithms on modern multicore and many-core architectures. Operations on adapted grids are characterized by irregular memory accesses and unbalanced compute patterns, which can be detrimental for performance. These challenges can at least be partially mitigated through careful analysis and adaptation of the algorithm and its implementation.

The main complications of 3D simulations with respect to 2D are the corresponding increases in memory footprint and computational operations. These challenges are often used to justify massively parallel distributed-memory implementations, despite the associated losses in computational efficiency. However, with an effectively adapted grid the memory footprint of most of the 3D simulations in this thesis would easily fit on $\mathcal{O}(10)$ modern compute nodes. The grid adaptation also reduces the number of compute operations, and the use of modern accelerators such as GPUs can further lower the time to solution. This means a high-performance multiresolution flow solver has the potential to bring 3D simulations of relevant scientific and engineering flows within the reach of hardware found in modern workstations, rather than supercomputing clusters.

1.3 Contributions

In this section we summarize the contributions that are made in this thesis towards overcoming the challenges presented above.

High Reynolds number vortex dynamics

- The validity of the remeshed Vortex Method for vortical flows is demonstrated by comparing with the established pseudo-spectral method for Reynolds numbers up to 10^4. We show that the accuracy of the rVM results can be greatly improved by using the six-point M_6^* interpolation kernel (Bergdorf 2007) with respect to the widely used four-point M_4' interpolation kernel (Monaghan 1985).

- The state of the art for the detailed study of vortex reconnection is advanced by simulating the flow at Reynolds number 10^4 on 4096 processors, both with and without initial axial flow. For the first time we show a secondary reconnection at late times, and explain how an axial flow within the tubes arises during the primary reconnection process.

FSI and optimization of self-propelled swimmers

- We extend the rVM to handle fluid-solid interaction, enabling simulations of multiple 3D deforming self-propelled swimmers. The solver is then coupled with the stochastic evolutionary optimization algorithm CMA-ES (Hansen, S. D. Müller, and Koumoutsakos 2003), to create a unique framework for the optimization of 3D self-propelled swimmers. This framework is designed to be generic and can simultaneously handle multiple fixed and deforming obstacles of arbitrary complexity.

- The escape mechanism of larval fish is optimized and it is shown that C-starts observed in natural swimmers are optimal. Analysis of the vorticity and velocity fields shows that the C-start mechanism relies on the swimmer to capture and accelerate a large region of fluid during a two-stage process of preparation and propulsion.

- Optimal shapes of anguilliform swimmers are found that maximize either speed or efficiency. The resulting shapes share some features with natural swimmers but provide better performance than natural larval fish shapes, albeit while operating with higher energetic requirements.

- We optimize both shape and body kinematics simultaneously to find 3D self-propelled swimmers with maximum speed or efficiency. The fastest morphokinematic solution consists of a slender profile performing a C-start type kinematics, showing that C-starts provide maximum acceleration also during steady-state swimming. Instead, the most efficient solution combines kinematics reminiscent of carangiform or subcarangiform swimming with a voluminous morphology.

Multiresolution flow simulations

- Based on an existing prototype of a 3D wavelet-based adaptive rVM (Rossinelli 2011) we develop and validate a multiresolution flow solver to efficiently simulate FSI problems, in particular 3D self-propelled swimmers.
- A new velocity solver for the 3D multiresolution flow solver is developed based on the multipole algorithm. The solver is implemented for shared-memory nodes with hybrid CPU/GPU support, and a basic extension to distributed memory is provided.
- We validate the solver and present for the first time a simulation of multiple 3D self-propelled swimmers based on the multiresolution framework, executed on only four compute nodes.

1.4 Thesis structure

This thesis is organized in four parts. In Part I we consider vortex dynamics simulations with the rVM, in the absence of solid obstacles, using a uniform-resolution distributed-memory code. Chapter 2 presents the equations governing the evolution of fluid flows. It then proceeds with introducing particle methods to solve transport problems, and how they can be improved by using frequent remeshing on an Eulerian grid. The remeshed particle methods are applied to the Navier-Stokes equations resulting in the remeshed Vortex Method. The chapter ends with a short introduction to the PPM software used for the simulations in both part I and part II. Subsequently, we show in chapter 3 the results of a comparison study between the established pseudo-spectral method and the rVM. We compare the results both for the Taylor-Green vortex at Reynolds number 1600 as for the case of vortex tube reconnection at Reynolds number 10^4. This latter flow case is then investigated more rigorously to uncover the dominating vortex dynamics in chapter 4.

In Part II we present our results relating to fluid-solid interaction and the optimization of self-propelled swimmers. Chapter 5 presents the extension of the rVM to incorporate two-way fluid structure interaction, and its coupling to an optimization technique. The next three chapters show results obtained with the resulting framework, starting with the optimization of escape mechanisms (chapter 6) for larval zebrafish. We then present an optimization of the morphology of anguilliform swimmers for speed and efficiency in chapter 7. We close this part with a full morphokinematic optimization, again for speed and efficiency, of undulatory swimmers in chapter 8.

Part III presents our efforts towards a 3D multiresolution rVM-FSI solver. In chapter 9 we explain the wavelet-basis for multiresolution analysis, and how this can be used to create a multiresolution rVM. We present details on the software and its optimizations for modern compute hardware. Chapter 10 presents our hybrid CPU/GPU 3D multipole-based solver optimized for block-structured multiresolution grids. The following two chapters show results obtained with the multiresolution flows solvers in 2D and 3D, respectively. Chapter 11 shows results of a study on the self-propulsion of a 2D counter-rotating cylinder pair. In chapter 12 we present validation, performance and results of the 3D multiresolution rVM flow solver for single and multiple swimmers.

In part IV we draw the conclusions from this work by summarizing the main contributions and presenting a perspective on possible future research directions.

The appendix in part V contains a gallery with some of the highlight images of our flow simulations, and supplementary details as referenced in the main text.

Part I

Vortex dynamics

CHAPTER 2

Equations and methods

In this chapter we present the equations and numerical methods for incompressible flow in a viscous fluid in the absence of solid bodies. Section 2.1 presents the Navier-Stokes equations in velocity-pressure and vorticity-velocity form. In section 2.2 we present how particle methods can be used to discretize transport problems, and how their convergence can be ensured with remeshing. The last section combines the Navier-Stokes equations with remeshed particle methods into the remeshed Vortex Method (rVM). We discuss the algorithm and the software we use for the uniform resolution simulations in this and the following part of this thesis.

2.1 Navier-Stokes equations

Throughout this thesis we consider the continuum representation of an incompressible flow in a viscous fluid with constant unit density. From the conservation of mass in an incompressible flow it follows that the velocity field (**u**) is solenoidal

$$\nabla \cdot \mathbf{u} = 0. \tag{2.1}$$

Conservation of momentum is governed by the Navier-Stokes equations, which in velocity-pressure formulation read

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u}, \tag{2.2}$$

where p is the pressure, ρ is the density and ν is the kinematic viscosity, defined as $\nu = \mu/\rho$ with μ the dynamic viscosity.

Taking the curl of the Navier-Stokes equations, and using the definition of the vorticity

$$\boldsymbol{\omega} = \nabla \times \mathbf{u}, \tag{2.3}$$

and the incompressibility condition (2.1), one arrives at the velocity-vorticity formulation of the Navier-Stokes equations

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu \Delta \boldsymbol{\omega}. \tag{2.4}$$

This is an evolution equation for the vorticity field. Its left-hand side represents the transport of vorticity from an Eulerian point of view. The right-hand side terms correspond respectively to stretching and diffusion.

It follows from equation (2.3) that the vorticity is also divergence-free. In the vorticity-velocity formulation we typically introduce a vector streamfunction $\boldsymbol{\Psi}$ that is related to the velocity by

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi}, \tag{2.5}$$

so that the velocity field is guaranteed to be divergence-free. To uniquely define the streamfunction we set $\nabla \cdot \boldsymbol{\Psi} = 0$ and then it follows from equations (2.3) and (2.5) that

$$\Delta \boldsymbol{\Psi} = -\boldsymbol{\omega}. \tag{2.6}$$

Boundary conditions used here are either periodic, or represent the free-space by ensuring that

$$\mathbf{u} \rightarrow \mathbf{U}_\infty \quad \text{as} \quad |\mathbf{x}| \rightarrow \infty, \tag{2.7}$$

where $\mathbf{U}_\infty$ is the free-stream velocity.

2.2 Particle methods for transport problems

We show how particle methods can be used to discretize transport problems. The extension to the full Navier-Stokes equations is presented in section 2.3.

We proceed by detailing how a function can be discretized with particles, and extend this to the solution of conservation laws. Then we highlight the importance of remeshing and analyze the two interpolation kernels used extensively in this thesis. Finally we show how this method, and the kernels in particular, can be written as finite-difference methods and extended with diffusion effects.

2.2.1 Particle function representation

It was shown in Raviart (1985) and Cottet and Koumoutsakos (2000) how we can construct a discrete representation of a continuous function using particles in $\mathbb{R}^d$. As starting point of the particle approximation we consider the integral identity

$$q(\mathbf{x}) = \int \delta(\mathbf{x}-\mathbf{y})q(\mathbf{y})\,\mathrm{d}\mathbf{y}, \tag{2.8}$$

where δ is the Dirac delta measure. We can discretize this identity using point particles according to the quadrature

$$q^h(\mathbf{x}) = \sum_p Q_p \delta(\mathbf{x}-\mathbf{x}_p), \tag{2.9}$$

where Q_p are the particle weights and $\mathbf{x}_p$ are the particle positions. The weights can be approximated using the midpoint quadrature rule

$$Q_p = \int_{v_p} q(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx q(\mathbf{x}_p)h^d = q_p h^d, \tag{2.10}$$

where v_p is the particle volume and h is the inter-particle distance.

To ensure our particles have finite size we replace the Dirac delta measure in equation (2.8) by an appropriately constructed mollification function ζ so that

$$q_\epsilon(\mathbf{x}) = q \star \zeta_\epsilon = \int \zeta_\epsilon(\mathbf{x}-\mathbf{y})q(\mathbf{y})\,\mathrm{d}\mathbf{y}, \tag{2.11}$$

where ϵ is the characteristic length of the kernel function and

$$\zeta_\epsilon(x) = \epsilon^{-d}\zeta\left(\frac{x}{\epsilon}\right). \tag{2.12}$$

This approximation introduces an error of size (Raviart 1985)

$$||q - q_\epsilon||_{L^\infty} \leq C_1 \epsilon^r \left\| \frac{\partial^r q}{\partial \mathbf{x}^r} \right\|_{L^\infty}, \tag{2.13}$$

given that the mollified kernel function satisfies

$$\int \zeta(x)\,\mathrm{d}x = 1, \tag{2.14}$$

$$\int x^\alpha \zeta(x)\,\mathrm{d}x = 0, \quad 1 \leq \alpha < r, \tag{2.15}$$

$$\int |x|^r |\zeta(x)|\,\mathrm{d}x < \infty. \tag{2.16}$$

The smooth function approximation (2.11) can then be discretized using again the midpoint quadrature rule to obtain

$$q_\epsilon^h(\mathbf{x}) = \sum_p Q_p \zeta_\epsilon(\mathbf{x} - \mathbf{x}_p) = \sum_p q_p h^d \zeta_\epsilon(\mathbf{x} - \mathbf{x}_p). \tag{2.17}$$

The error of this discretization, following a midpoint rule, for sufficiently smooth ζ, is

$$\|q_\epsilon - q_\epsilon^h\|_{L^\infty} \leq C_2 \left(\frac{h}{\epsilon}\right)^m \left\|\frac{\partial^m q}{\partial \mathbf{x}^m}\right\|_{L^\infty}, \tag{2.18}$$

where $m \geq 1$ depends on the smoothness properties of ζ. It follows that for convergence, we have to ensure $(h/\epsilon) < 1$, which implies the particles must overlap.

2.2.2 Particle methods for conservation laws

Let us now consider the following conservation law

$$\frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{u}q) = 0, \tag{2.19}$$

$$q(\mathbf{x}, 0) = q_0(\mathbf{x}), \tag{2.20}$$

Integrating this over a material volume $V(t)$ gives

$$\int_{V(t)} \frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{u}q)\,\mathrm{d}V = \frac{\mathrm{d}}{\mathrm{d}t} \int_{V(t)} q\,\mathrm{d}V = 0. \tag{2.21}$$

The conservation law therefore is equivalent to conservation of the integral quantity $\int q\,\mathrm{d}V$ in the flow field $\mathbf{u}$.

The point particle method solution to this set of equations is given by

$$q^h(\mathbf{x}, t) = \sum_p Q_p \delta(\mathbf{x} - \mathbf{X}_p(t)), \tag{2.22}$$

$$\mathbf{X}_p(t) = \mathbf{X}(t; \mathbf{x}_p, 0), \tag{2.23}$$

$$Q_p = h^d q_0(\mathbf{x}_p). \tag{2.24}$$

where Q_p are the weights obtained from the particle discretization of the initial condition (Raviart 1985).

When using a smooth kernel function, either with compact support or using an initial condition that decays sufficiently rapidly, equation (2.22) changes to

$$q_\epsilon^h(\mathbf{x}, t) = \sum_p Q_p \zeta_\epsilon(\mathbf{x} - \mathbf{X}_p(t)). \tag{2.25}$$

The error of this approximation is bounded by a combination of the quadrature error (2.18) and mollification error (2.13)

$$||q - q_\epsilon^h||_{L^\infty} \leq C_1 \epsilon^r \left|\left|\frac{\partial^r q_0}{\partial \mathbf{x}^r}\right|\right| + C_2 \left(\frac{h}{\epsilon}\right)^m \left|\left|\frac{\partial^m q_0}{\partial \mathbf{x}^m}\right|\right|. \tag{2.26}$$

In practice, we initialize the particles according equation (2.24) and solve the following set of ODEs

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} &= \mathbf{u}(\mathbf{x}_p, t), \\ \frac{\mathrm{d}Q_p}{\mathrm{d}t} &= 0. \end{aligned} \tag{2.27}$$

The approximation to $q(\mathbf{x}, t)$ can then be reconstructed according to

$$q_\epsilon^h(\mathbf{x}, t) = \sum_p Q_p \zeta_\epsilon(\mathbf{x} - \mathbf{x}_p). \tag{2.28}$$

2.2.3 Remeshing

To ensure the particle overlap that is essential for convergence, the particles are remeshed onto a uniform Cartesian grid at the end of every time step (Koumoutsakos 1997; Cottet and Koumoutsakos 2000). The remeshing involves a particle-to-mesh (P2M) interpolation of the particle weights followed by resetting the particle positions to the grid point locations. To interpolate quantities from the grid back onto the particles we perform a mesh-to-particle (M2P) interpolation.

Mathematically, the P2M interpolation step is a discrete convolution between a kernel and the particle weights

$$q_\mathbf{i} = \sum_p q_p W\left(\frac{\mathbf{x}_p - \mathbf{x}_\mathbf{i}}{h}\right), \tag{2.29}$$

where $\mathbf{i}$ denotes the d-dimensional grid point index, $\mathbf{x}_\mathbf{i} = h\mathbf{i}$ with h the constant uniform grid spacing, and $W(\mathbf{x})$ is the interpolation kernel. The operation can be inverted to obtain the M2P interpolation

$$q_p = \sum_\mathbf{i} q_\mathbf{i} W\left(\frac{\mathbf{x}_\mathbf{i} - \mathbf{x}_p}{h}\right). \tag{2.30}$$

The interpolation kernel $W(\mathbf{x})$ in $d > 1$ dimensions is typically constructed from the tensorial product of a one-dimensional function

$$W(\mathbf{x}) = \prod_{n=1}^{d} M(\{\mathbf{x}\}_n). \tag{2.31}$$

The interpolation kernel $M(x)$ is typically designed to conserve the first r moments of the field it is interpolating

$$\sum_{\mathbf{i}} q_{\mathrm{i}}(\mathbf{x}_{\mathrm{i}} - \mathbf{x})^{\alpha} = \sum_{p} q_p(\mathbf{x}_p - \mathbf{x})^{\alpha} \quad 0 \leq \alpha < r, \tag{2.32}$$

where $\mathbf{x}$ is an arbitrary location. This condition is equivalent to reproduction of polynomials up to degree $r-1$, which can be shown by first combining equations (2.29) and (2.32)

$$\sum_{\mathbf{i}} \sum_{p} q_p W\left(\frac{\mathbf{x}_{\mathrm{i}} - \mathbf{x}_p}{h}\right)(\mathbf{x}_{\mathrm{i}} - \mathbf{x})^{\alpha} = \sum_{p} q_p(\mathbf{x}_p - \mathbf{x})^{\alpha} \quad 0 \leq \alpha < r. \tag{2.33}$$

Now we use the binomial theorem to expand the terms involving α on both the left and right hand side and rearrange to obtain

$$\sum_{p} q_p \sum_{k=0}^{\alpha} \left(\sum_{\mathbf{i}} W\left(\frac{\mathbf{x}_p - \mathbf{x}_{\mathrm{i}}}{h}\right) \mathbf{x}_i^k \right) \binom{\alpha}{k} (-\mathbf{x})^{\alpha-k} = \sum_{p} q_p \sum_{k=0}^{\alpha} \mathbf{x}_p^k \binom{\alpha}{k} (-\mathbf{x})^{\alpha-k}, \tag{2.34}$$

or

$$\sum_{\mathbf{i}} W\left(\frac{\mathbf{x}_p - \mathbf{x}_{\mathrm{i}}}{h}\right) \mathbf{x}_{\mathrm{i}}^{\alpha} = \mathbf{x}_p^{\alpha} \quad 0 \leq \alpha < r. \tag{2.35}$$

A commonly used kernel in particle methods is the M_4' kernel (Monaghan 1985), defined as

$$M_4'(x) = \begin{cases} 0 & \text{if } |x| > 2, \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 < |x| \leq 2, \\ 1 - \frac{5}{2}x^2 + \frac{3}{2}|x|^3 & \text{if } |x| \leq 1, \end{cases} \tag{2.36}$$

and plotted in figure 2.1. This kernel has a four-point support and conserves the first $r = 3$ moments of the field, and therefore will reproduce up to quadratic polynomial functions exactly, making its error $\mathcal{O}(h^3)$. Its Fourier transform and series expansion around $k = 0$ are given by

$$\int_{-\infty}^{\infty} M_4'(x) e^{-2\pi i k x} \, \mathrm{d}x = \frac{\sin(k^3\pi^3)(3\sin(k\pi) - 2k\pi\cos(k\pi))}{k^4\pi^4} \tag{2.37}$$

$$= 1 - \frac{(k\pi)^4}{5} + \mathcal{O}(k^6). \tag{2.38}$$

We note that the $\mathcal{O}(k^4)$ term actually implies that the first *four* moments of this kernel will be zero (Cottet and Koumoutsakos 2000), so that in physical space

$$\sum_{i} W\left(\frac{\mathbf{x}_i}{h}\right) \mathbf{x}_i^{\alpha} = 0^{\alpha} \quad 0 \leq \alpha < r', \tag{2.39}$$

for $r' = 4$, even though above $r = 3$. The reason for this discrepancy is that in equation (2.39) we consider the moment expansion around $x = 0$, so that for symmetric kernels all odd-numbered moments are automatically conserved. This means that if equation (2.35) is satisfied up to $r = 3$, equation (2.39) will be satisfied up to $r' = 4$. Nevertheless following equation (2.35), we can not conserve more than the first three moments of the field that is interpolated.

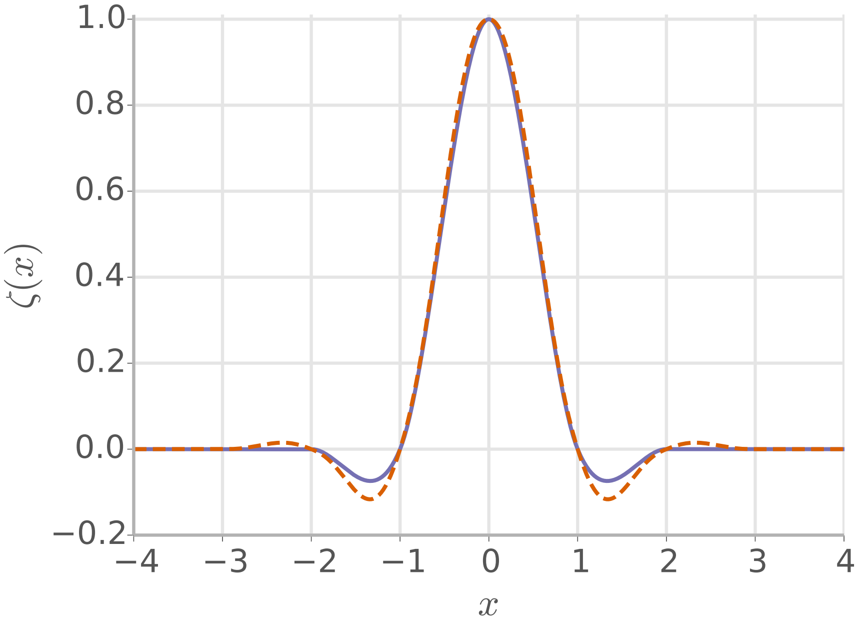


Figure 2.1: The four-point M_4' kernel (solid blue) and the six-point M_6^* kernel (dashed red) used for remeshing the particles onto the mesh in this work.

In addition to the M_4' kernel, we will use a kernel derived in (Bergdorf 2007), denoted as the M_6^* kernel. This kernel has a support of six points, conserves the first $r = 5$ moments of the field and reproduces quartic polynomials exactly. It is defined by

$$M_6^*(x) = \begin{cases} 0 & \text{if } |x| > 3, \\ -\frac{1}{24}(|x|-2)(|x|-3)^3(5|x|-8) & \text{if } 2 < |x| \leq 3, \\ \frac{1}{24}(|x|-1)(|x|-2)(25|x|^3 - 114x^2 + 153|x| - 48) & \text{if } 1 < |x| \leq 2, \\ -\frac{1}{12}(|x|-1)(25x^4 - 38|x|^3 - 3x^2 + 12|x| + 12) & \text{if } |x| \leq 1, \end{cases} \tag{2.40}$$

and shown in figure 2.1. The Fourier transform of the M_6^* kernel, and its series expansion around $k = 0$ are

$$\int_{-\infty}^{\infty} M_6^*(x)e^{-2\pi ikx}\,\mathrm{d}x = \frac{\sin(k^5\pi^5)\big((25-7k^2\pi^2)\sin(k\pi)-24k\pi\cos(k\pi)\big)}{k^6\pi^6} \tag{2.41}$$

$$= 1 - \frac{143(k\pi)^6}{945} + \mathcal{O}(k^8). \tag{2.42}$$

Again, following the above discussion and since the first non-conserved moment is odd, this kernel will satisfy equation (2.39) for $r' = 6$, consistent with the $\mathcal{O}(k^6)$ in equation (2.42). More details on this theory and different kernels can be found in Cottet and Koumoutsakos (2000) and Cottet, Etancelin, et al. (2014).

2.2.4 Lagrangian time step criterion

A Lagrangian treatment of the convection term removes the Courant-Friedrichs-Levy (CFL) stability restriction that couples the time step to the grid spacing. Instead, to guarantee particle overlap, the remeshed particle methods are solely bound by the velocity gradient tensor. Theoretically, only the strain rate tensor results in deformation in the flow. However, curved particle paths typical of purely rotational flows can also result in the loss of overlap, due to time integration errors. Therefore typically the timestep is to be restricted by all components of the velocity gradient tensor, so that

$$\delta t < C\left(||\nabla\mathbf{u}||_\infty\right)^{-1}, \tag{2.43}$$

where C is a constant that depends only on the characteristic size of the kernel. Depending on accuracy and performance considerations, we can then choose a Lagrangian CFL (LCFL) smaller than C, so that

$$\delta t||\nabla\mathbf{u}||_\infty = \mathrm{LCFL}. \tag{2.44}$$

Depending on the flow and the numerical setup, an LCFL-based time step can be much more relaxed than a CFL-based time step. In the obvious case of linear advection or rigid-body motions, the flow is deformation-free and the time step for remeshed particle methods can be arbitrarily large. In general this advantage is retained for convection-dominated flows. On the other hand, high Reynolds number turbulence or wall-bounded flows with thin boundary layers are characterized by large velocity gradients rather than large velocity magnitudes. The advantages of particle methods with respect to grid-based methods will therefore differ between specific flows.

2.2.5 Remeshing and finite difference methods

The equivalence between remeshed particle methods and finite difference operators was shown in Cottet and Weynans (2006). Here we follow their analysis to show this equivalence for both of the kernels discussed above, in the case of 1D linear advection in a periodic domain:

$$\frac{\partial q}{\partial t} + U\frac{\partial q}{\partial x} = 0, \tag{2.45}$$

$$q(x,0) = q_0(x), \tag{2.46}$$

where $U > 0$ is constant in space and time. In Fourier space, the exact solution and its expansion in k can be written as

$$\hat{q}(k,t) = \hat{q}_0 e^{-ikUt} \tag{2.47}$$

$$\begin{aligned}&= \hat{q}_0\bigg(1 - ik(Ut) - \frac{1}{2}k^2(Ut)^2 + \frac{1}{6}ik^3(Ut)^3 + \frac{1}{24}k^4(Ut)^4 \\ &\quad - \frac{1}{120}ik^5(Ut)^5 + \mathcal{O}(k^6)\bigg).\end{aligned} \tag{2.48}$$

If we apply the remeshed particle method to approximate the solution of $q(x)$ at grid point i and time step $n+1$ given the values at time step n we get

$$q_i^{n+1} = \sum_i q_i^n W\left(\frac{U\delta t}{h}\right) = \sum_i q_i^n W(\lambda), \tag{2.49}$$

where we introduced $\lambda = U\delta t/h$. Substituting the generic kernel $W(\lambda)$ for the M_4' kernel discussed above we find, for $0 \leq \lambda < 1$

$$q_i^{n+1} = \gamma_1 q_{i+1}^n + \gamma_0 q_i^n + \gamma_{-1} q_{i-1}^n + \gamma_{-2} q_{i-2}^n, \tag{2.50}$$

where the weights γ_i are given as

$$\gamma_1 = M_4'(\lambda+1) = -\frac{1}{2}\lambda + \lambda^2 - \frac{1}{2}\lambda^3, \tag{2.51}$$

$$\gamma_0 = M_4'(\lambda) = 1 - \frac{5}{2}\lambda^2 + \frac{3}{2}\lambda^3, \tag{2.52}$$

$$\gamma_{-1} = M_4'(1-\lambda) = \frac{1}{2}\lambda + 2\lambda^2 - \frac{3}{2}\lambda^3, \tag{2.53}$$

$$\gamma_{-2} = M_4'(2-\lambda) = -\frac{1}{2}\lambda^2 + \frac{1}{2}\lambda^3. \tag{2.54}$$

Substituting these weights into (2.50), taking the Fourier transform and performing a Taylor expansion on the result gives

$$\hat{q}_k^{n+1} = \hat{q}_k^n \left(\gamma_1 e^{ikh} + \gamma_0 + \gamma_{-1} e^{-ikh} + \gamma_{-2} e^{-2ikh}\right) \tag{2.55}$$

$$= \hat{q}_k^n \left(1 - ih\lambda k - \frac{1}{2}(h\lambda k)^2 + \frac{1}{6} i(kh)^3 \left(\lambda - 3\lambda^2 + 3\lambda^3\right) + \mathcal{O}(kh)^4\right). \tag{2.56}$$

Comparing equation (2.56) with the Taylor expansion for the exact solution we find an exact match up to $\mathcal{O}(kh)^3$. In physical space the finite-difference scheme corresponding to the M_4' kernel is obtained from expanding equation (2.50) and rearranging terms

$$\begin{aligned} q_i^{n+1} = q_i^n &- U\delta t \left(\frac{q_{i+1}^n - q_{i-1}^n}{2h}\right) \\ &+ \frac{(U\delta t)^2}{2}\left(\frac{2q_{i+1}^n - 5q_i^n + 4q_{i-1}^n - q_{i-2}^n}{h^2}\right) \\ &- \frac{(U\delta t)^3}{6}\left(\frac{3q_{i+1}^n - 9q_i^n + 9q_{i-1}^n - 3q_{i-2}^n}{h^3}\right). \end{aligned} \tag{2.57}$$

The same analysis for the M_6^* kernel gives the following update expression for q_i

$$q_i^{n+1} = \gamma_2 q_{i+2}^n + \gamma_1 q_{i+1}^n + \gamma_0 q_i^n + \gamma_{-1} q_{i-1}^n + \gamma_{-2} q_{i-2}^n + \gamma_{-3} q_{i-3}^n, \tag{2.58}$$

with the following weights

$$\gamma_2 = M_6^*(\lambda + 2) = \frac{1}{24}\left(2\lambda - \lambda^2 - 9\lambda^3 + 13\lambda^4 - 5\lambda^5\right), \tag{2.59}$$

$$\gamma_1 = M_6^*(\lambda + 1) = \frac{1}{24}\left(-16\lambda + 16\lambda^2 + 39\lambda^3 - 64\lambda^4 + 25\lambda^5\right), \tag{2.60}$$

$$\gamma_0 = M_6^*(\lambda) = \frac{1}{12}\left(12 - 15\lambda^2 - 35\lambda^3 + 63\lambda^4 - 25\lambda^5\right), \tag{2.61}$$

$$\gamma_{-1} = M_6^*(1 - \lambda) = \frac{1}{12}\left(8\lambda + 8\lambda^2 + 33\lambda^3 - 62\lambda^4 + 25\lambda^5\right), \tag{2.62}$$

$$\gamma_{-2} = M_6^*(2 - \lambda) = \frac{1}{24}\left(-2\lambda - \lambda^2 - 33\lambda^3 + 61\lambda^4 - 25\lambda^5\right), \tag{2.63}$$

$$\gamma_{-3} = M_6^*(3 - \lambda) = \frac{1}{24}\left(7\lambda^3 - 12\lambda^4 + 5\lambda^5\right). \tag{2.64}$$

Substution into the Fourier transform of equation (2.58) and expanding gives

$$\hat{q}_k^{n+1} = \hat{q}_k^n \left(\gamma_2 e^{2ikh} + \gamma_1 e^{ikh} + \gamma_0 + \gamma_{-1} e^{-ikh} + \gamma_{-2} e^{-2ikh} + \gamma_{-3} e^{-3ikh}\right) \tag{2.65}$$

$$\begin{aligned} = \hat{q}_k^n \Big(& 1 - ih\lambda k - \frac{1}{2}(h\lambda k)^2 + \frac{1}{6} i(h\lambda k)^3 + \frac{1}{24}(h\lambda k)^4 \\ & - \frac{1}{120} i(kh)^5 \left(-4\lambda + 40\lambda^3 - 60\lambda^4 + 25\lambda^5\right) + \mathcal{O}(kh)^6\Big). \end{aligned} \tag{2.66}$$

Comparing equation (2.66) with the Taylor expansion for the exact solution we find an exact match up to $\mathcal{O}(kh)^5$. Again, we can also substitute the weights in the expression in physical space to obtain the equivalent finite-difference scheme

$$\begin{aligned} q_i^{n+1} =& q_i^n - U\delta t\left(\frac{-q^n_{i+2}+8q^n_{i+1}-8q^n_{i-1}+q^n_{i-2}}{12h}\right)+ \\ & \frac{(U\delta t)^2}{2}\left(\frac{-q^n_{i+2}+16q^n_{i+1}-30q^n_i+16q^n_{i-1}-q^n_{i-2}}{12h^2}\right)- \\ & \frac{(U\delta t)^3}{6}\left(\frac{9q^n_{i+2}-39q^n_{i+1}+70q^n_i-66q^n_{i-1}+33q^n_{i-2}-7q^n_{i-3}}{4h^3}\right)+ \\ & \frac{(U\delta t)^4}{24}\left(\frac{13q^n_{i+2}-64q^n_{i+1}+126q^n_i-124q^n_{i-1}+61q^n_{i-2}-12q^n_{i-3}}{h^4}\right)- \\ & \frac{(U\delta t)^5}{120}\left(25\frac{q^n_{i+2}-5q^n_{i+1}+10q^n_i-10q^n_{i-1}+5q^n_{i-2}-q^n_{i-3}}{h^5}\right). \end{aligned} \tag{2.67}$$

These analyses show the equivalent finite difference scheme for the M_4' and M_6^* kernels for the case of 1D linear advection with $0 \leq \text{CFL} < 1$. The extensions to non-linear convection and higher CFL-numbers is discussed in Cottet and Weynans (2006), although the specifics for the kernels considered here are still open. Furthermore, as shown in Magni and Cottet (2012), in case of non-linear convection the accuracy of remeshed particle methods can reduce to first order when the local CFL-number crosses an integer value between two neighboring particles. In practice we have not observed any significant effects of this phenomenon on the results of the incompressible flow simulations performed in this thesis.

2.2.6 Remeshing and diffusion

It was shown in Wee and Ghoniem (2006) that by changing the weights of the interpolation kernel, one can incorporate the effects of a diffusion term to the updated solution. Here we show how the above M_4' and M_6^* kernels can be modified to take this into account. The analysis for M_4' has been done by Prof. Anthony Leonard (personal communication), and we repeat here his results and the corresponding analysis for the M_6^* kernel. In the following, we solve

$$\frac{\partial q}{\partial t} + U\frac{\partial q}{\partial x} = \nu\frac{\partial^2 q}{\partial x^2}, \tag{2.68}$$

$$q(x,0) = q_0(x), \tag{2.69}$$

where ν is a constant representing the viscosity of the fluid. The exact solution of this equation in Fourier space is

$$
\begin{aligned}
\hat{q}(k,t) &= \hat{q}_0 e^{-ikUt-\nu k^2 t} && (2.70)\\
&= \hat{q}_0 \Big(1 - ik(Ut) - \frac{1}{2}k^2\left((Ut)^2 + 2\nu t\right) \\
&\quad + \frac{1}{6} ik^3 \left((Ut)^3 + 6(Ut)(\nu t)\right) + \frac{1}{24}k^4\left((Ut)^4 + 12(Ut)^2(\nu t) + 12(\nu t)^2\right) && (2.71)\\
&\quad - \frac{1}{120} ik^5\left((Ut)^5 + 20(Ut)^3(\nu t) + 20(Ut)(\nu t)^2\right) + \mathcal{O}(k^6)\Big).
\end{aligned}
$$

With the M_4' kernel, or any symmetric kernel with four-point support, the solution $\hat{q}(k,t)$ will be approximated by equation (2.55), where the γ_i weights are now updated to account for the $\nu \neq 0$ terms in equation (2.71). We can therefore seek to generalize the M_4' kernel into an $M_4'^D$ kernel that incorporates diffusion

$$
M_4'^D(x) = M_4'(x) + \begin{cases} 0 & \text{if } |x| > 2, \\ \nu\delta t(a_1 + b_1 x) & \text{if } 1 < |x| \leq 2, \\ \nu\delta t(a_2 + b_2 x) & \text{if } |x| \leq 1. \end{cases} \tag{2.72}
$$

The two additional terms proportional to $\nu\delta t$ and $\nu\delta t x$ are suggested by the $\nu \neq 0$ terms of orders k^2 and k^3 in equation (2.71). Remeshing with the $M_4'^D$ kernel then gives the following update rule

$$
\begin{aligned}
q_i^{n+1} &= \gamma_1^D q_{i+1}^n + \gamma_0^D q_i^n + \gamma_{-1}^D q_{i-1}^n + \gamma_{-2}^D q_{i-2}^n && (2.73)\\
&= M_4'^D(\lambda+1)q_{i+1}^n + M_4'^D(\lambda)q_i^n + M_4'^D(1-\lambda)q_{i-1}^n + M_4'^D(2-\lambda)q_{i-2}^n. && (2.74)
\end{aligned}
$$

We take the Fourier transform and solve for the unknown coefficients a_i and b_i by equating the first four orders of k with the exact solution (2.71), giving

$$
\begin{aligned}
a_1 &= 2, \quad b_1 = -1, && (2.75)\\
a_2 &= -2, \quad b_2 = 3. && (2.76)
\end{aligned}
$$

With these coefficients the additional terms in the modified $M_4'^D$ kernel match the first four terms in the exact solution of the diffusion equation. We note firstly that the analysis depends only on the support of the kernel, so that the a_i and b_i terms can be added to any four-point interpolation kernel to incorporate the effects of diffusion. Secondly, this kernel reproduces the first four terms of the exact solution whereas the proposed M_4'-based diffusion kernel of Wee and Ghoniem (2006) reproduces only the first three terms.

A similar analysis can be done for the M_6^* kernel. In this case we have a larger support and so we can aim for higher accuracy. In particular we wish to match the first six orders of k in the exact solution. Inspecting those terms equation (2.71) shows that we will need to append the M_6^* kernel as follows

$$M_6^{*D}(x) = M_6^*(x) + \begin{cases} 0 & \text{if } |x| > 3, \\ \nu\delta t\left(a_1 + b_1 x + c_1 x^2 + d_1 x^3 + e_1 \nu\delta t + f_1 \nu\delta t x\right) & \text{if } 2 < |x| \le 3, \\ \nu\delta t\left(a_2 + b_2 x + c_2 x^2 + d_2 x^3 + e_2 \nu\delta t + f_2 \nu\delta t x\right) & \text{if } 1 < |x| \le 2, \\ \nu\delta t\left(a_3 + b_3 x + c_3 x^2 + d_3 x^3 + e_3 \nu\delta t + f_3 \nu\delta t x\right) & \text{if } |x| \le 1. \end{cases} \tag{2.77}$$

This gives 18 unknowns whereas we only have 6 equations. However, as additional constraint we require that the values of our coefficients are independent of λ. This can be done for each order of λ and gives a unique solution

$$a_1 = 15/4, \quad b_1 = -17/4, \quad c_1 = 3/2, \quad d_1 = -1/6, \quad e_1 = 3/2, \quad f_1 = -1/2, \tag{2.78}$$

$$a_2 = -5/4, \quad b_2 = 25/4, \quad c_2 = -9/2, \quad d_2 = 5/6, \quad e_2 = -9/2, \quad f_2 = 5/2, \tag{2.79}$$

$$a_3 = -5/2, \quad b_3 = 5/2, \quad c_3 = 3, \quad d_3 = -5/3, \quad e_3 = 3, \quad f_3 = -5. \tag{2.80}$$

The M_6^{*D} interpolation kernel with these coefficients matches the first six orders of the exact solution of the diffusion equation. Again we note that the extra terms are general to any six-point interpolation kernel since we have not used any details of the original M_6^* kernel in this analysis.

In the practical flow applications presented in this thesis we solve the diffusion term by applying a finite-difference technique on the Eulerian grid (detailed in the next section), instead of using the kernels presented in this section. A comparison of the accuracy and performance of diffusion-enhanced interpolation kernels and finite-difference techniques in the context of 3D flow simulations is left for future work.

2.3 Remeshed vortex method

The remeshed Vortex Method (Koumoutsakos 1997; Koumoutsakos 2005) applies the framework of the remeshed particle method to the Navier-Stokes equations in vorticity-velocity formulation. The vorticity field is represented with a set of particles

$$\boldsymbol{\omega}(\mathbf{x}, t) \approx \sum_p \boldsymbol{\Gamma}_p(t)\zeta_\epsilon(\mathbf{x} - \mathbf{x}_p(t)), \tag{2.81}$$

where $\boldsymbol{\Gamma}_p(t)$ and $\mathbf{x}_p(t)$ denote the strength and position, respectively, of the p-th particle at time t.

2.3.1 Algorithm

Applying the particle representation (2.81) to the Navier-Stokes equations results in a set of ordinary differential equations (ODEs) for the particle strengths and the particle positions

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_p}{\mathrm{d}t} = h^d(\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\Delta\boldsymbol{\Gamma}_p, \tag{2.82}$$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}_p, t). \tag{2.83}$$

These differential equations are evolved using a suitable time integration technique, and at the end of every time step the particles are remeshed onto a uniform Cartesian grid. The corresponding algorithm for one time step, or the sub-step of a multi-step time integration scheme, is given in algorithm 1. Here the subscript p represents particle values, otherwise the variables are represented on the grid. The operators $\mathrm{P2M}(\cdot, \mathbf{x}_p)$ and $\mathrm{M2P}(\cdot, \mathbf{x}_p)$ represent the particle-to-mesh and mesh-to-particle operations as defined in equations (2.29) and (2.30), respectively. We denote the right-hand side of the Navier-Stokes equations by $\delta\boldsymbol{\omega}$.

Algorithm 1 One (sub)time step of the remeshed vortex method

$$\Delta\boldsymbol{\Psi} = -\boldsymbol{\omega} \tag{2.84}$$

$$\mathbf{u} = \nabla\times\boldsymbol{\Psi} \tag{2.85}$$

$$\delta\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\Delta\boldsymbol{\omega} \tag{2.86}$$

$$\mathbf{u}_p = \mathrm{M2P}(\mathbf{u}, \mathbf{x}_p) \tag{2.87}$$

$$(\delta\boldsymbol{\omega})_p = \mathrm{M2P}(\delta\boldsymbol{\omega}, \mathbf{x}_p) \tag{2.88}$$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}_p \tag{2.89}$$

$$\frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = \delta\boldsymbol{\omega}(\mathbf{x}_p) \tag{2.90}$$

$$\boldsymbol{\omega} = \mathrm{P2M}(\boldsymbol{\omega}_p, \mathbf{x}_p) \tag{2.91}$$

For uniform resolution grids, as considered in this part of the thesis, equations (2.84) and (2.85) are solved together in Fourier space, using three forward and three inverse Fast Fourier Transforms (FFTs) to transform each component of the vector field. The differential operators in the right-hand side of equation (2.86) are approximated using fourth-order central finite difference methods. The P2M and M2P operations are done with either of the M_4' and M_6^* interpolation kernels. Time integration for the simulations performed in this part of the thesis is done using third or fourth-order Runge-Kutta schemes. A study on the accuracy of the different interpolation kernels and time integration techniques for practical flow problems is given in chapter 3.

2.3.2 Boundary conditions

In this work the boundary conditions on the computational domain are either free-space or periodic. In the periodic case, the particles that exit the domain re-enter from the other side, and ghost values on the grid are translated copies from the opposite face of the domain. In case free-space boundary conditions are used, the FFT fields are appended with zeros to eliminate the effects of aliasing as described in Hockney and Eastwood (1981) and Chatelain and Koumoutsakos (2010). In this case, the vorticity field will be completely contained within the computational domain, and so any particles that exit the domain are ensured to carry negligible vorticity and can be removed without affecting the simulation.

2.3.3 Divergence

Equations (2.1) and (2.3) state that both the velocity and the vorticity field remain divergence-free at all times. By construction, the velocity field is re-computed in a divergence-free manner at each time step according to equation (2.85). Although the M2P operation (2.87) could introduce some spurious divergence in the velocity field, this will not be carried along at subsequent time steps, and in practice does not present problems.

The 3D vorticity field, on the under hand, is not guaranteed to be divergence-free at the beginning of each time step. In fact, although the right-hand side of the vorticity evolution equation is divergence-free, spurious divergence due to spatial and temporal discretization errors, as well as the P2M operation (2.91) could accumulate during a simulation. This in turn can cause the simulation to behave erratically and, in the worst case, become unstable. To prevent this we periodically perform a solenoidal projection of the vorticity field. We start by considering the Helmholtz decomposition of a vorticity field $\boldsymbol{\omega}'$ that contains a spurious non-solenoidal component

$$\boldsymbol{\omega}' = \nabla \times \mathbf{A} + \nabla B. \tag{2.92}$$

Taking the divergence of this equation produces a scalar Poisson equation for B

$$\Delta B = \nabla \cdot \boldsymbol{\omega}'. \tag{2.93}$$

We solve this equation in Fourier space by means of the FFT. With B known, we can perform the projection

$$\boldsymbol{\omega} = \boldsymbol{\omega}' - \nabla B, \tag{2.94}$$

so that our resulting vorticity field $\boldsymbol{\omega}$ is indeed divergence-free.

A drawback of this approach is the introduction of a new parameter, the frequency of reprojection. Typically we perform one projection every 5 to 25 time steps. Another drawback is that this approach will adjust every component of the vorticity vector irrespective of their original values. This can create spurious vorticity, for instance in case the original field consists of only two non-zero vector components, such as for a planar vortex ring. However the corrections are typically small and we have not observed a negative impact of these drawbacks on our results.

2.3.4 Software

For all uniform resolution simulations considered in this thesis we use an rVM-based flow solver implemented for massively parallel computing architectures in the Parallel Particle-Mesh (PPM) library (Sbalzarini et al. 2006). This library offers an abstraction layer over the Message-Passing Interface (MPI) for particle-mesh simulations. It implements the definition of topologies (subdomains and their assignment to processors), load-balancing, communications, and particle-mesh operations such as interpolation. PPM is written in Fortran 90 on top of MPI.

The rVM-based flow solver uses parallel 3D FFTs implemented using the topologies of the PPM library, calling 1D FFT-routines based on the FFTW library (Frigo and S. Johnson 1998). The particle/mesh portions of our code (particle advection, particle mapping, mesh ghosting, particle-to-mesh and mesh-to-particle interpolation) use a Cartesian communicator and non-blocking point-to-point communications. Scaling of the code has been demonstrated for up to 16 384 processors (Chatelain, Curioni, et al. 2008). The strongest limitations on weak scaling are the communication steps between the 1D pencil topologies that are required for the FFTs. Before this thesis, the M_4' interpolation kernel and a low-storage third-order Runge-Kutta time-stepping scheme (J. Williamson 1980) were implemented and used for simulations. In this configuration, this solver has successfully been applied to several three-dimensional vortical flows, such as the decay of vortex rings (Bergdorf, Koumoutsakos, and Leonard 2007) and the destruction of aircraft wakes (Chatelain, Gazzola, et al. 2011).

For free-space boundary conditions the flow solver allows for dynamic adaptation of the domain size to accommodate a growing or shrinking support of the vorticity field.

Specifically, in each of the three directions the first and last few grid points of the domain box are designated as a buffer zone. Whenever the vorticity enters one of these buffer zones the domain is extended in that dimension. Similarly, whenever the vorticity leaves a buffer zone the domain size in that direction is reduced. In this way the domain always exclusively covers the support of the vorticity field and dynamically tracks it during the simulation. In the current part of this thesis we only use periodic boundary conditions, while in part II the simulations rely extensively on free-space boundary conditions and dynamic domain adaptation.

CHAPTER 3

A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers

In chapter 2 we considered the use of particle methods to solve transport equations and detailed the remeshing process to ensure convergence. We showed how the particle-mesh framework can be used to discretize the 3D Navier-Stokes equations, resulting in the remeshed vortex method (rVM).

Here we aim to establish the accuracy of the rVM results for high Reynolds number 3D vortex dynamics. We present a comparison between the rVM and an established pseudo-spectral method for two vortical flow applications. The first case is the evolution of a 3D Taylor-Green vortex at circulation-based Reynolds number $Re_\Gamma = 1600$, for which reference results are available in literature. For the second test case we consider the reconnection between two vortex tubes at $Re_\Gamma = 10\,000$. We investigate how the interpolation kernel and time stepping scheme change the performance of the rVM compared with the pseudo-spectral method.

The current chapter is based on van Rees, Leonard, et al. (2011).

3.1 Background

Vortex methods are arguably the first numerical method used for the simulation of vortical flows starting with the hand-calculations of Rosenhead in the beginning of last century (Rosenhead 1931). Vortex methods were considered as the method of choice for external flows with compact vorticity (Chorin 1973) due to their low numerical dissipation and they were among the first techniques used for simulations of 3D vortical flows (Leonard 1980; Anderson and C. Greengard 1985; Leonard 1985). Later it was realized (Koumoutsakos 1997) that the accuracy of the method hinges on the use of a regularization procedure to remedy the inaccuracies due to the distortion of the computational elements which follows from their Lagrangian adaptivity. In the remeshed vortex method (rVM) (Cottet and Koumoutsakos 2000; Winckelmans 2004; Koumoutsakos 2005), Lagrangian vortex particles are used to simulate the convective part of the equations and particles are mapped onto grid nodes at each time step so as to ensure the convergence of the method and to compute efficiently the solution of the Poisson equation that determines their velocity. This gives the rVM some inherent advantages over other methods, such as its adaptivity and the lack of a CFL restriction on the time step, which allows large time steps during the simulation. It is important to note that the use of a grid based solver for the Poisson equation accommodates a wide range of boundary conditions that may not be possible when using techniques such as the Fast Multipole Method (L. Greengard and Rokhlin 1987). We are interesting in exploiting these advantages for high Reynolds number vortical flows while retaining the accuracy of the results. The method has already successfully been used to perform direct numerical simulations up to $Re_\Gamma = 7500$ (Bergdorf, Koumoutsakos, and Leonard 2007; Chatelain, Curioni, et al. 2008). To this end, we will compare our remeshed vortex method with a pseudo-spectral method. The pseudo-spectral method is well suited for the simulation of high Reynolds number flows in simple domains and can be considered as a reference method. The goal of the current study is to validate the vortex method as an accurate and fast alternative to the pseudo-spectral method for high Reynolds number flow cases.

The first study of comparing the vortex method with the pseudo-spectral method was undertaken by (Cottet, Michaux, et al. 2002). In that study, the comparison was performed for isotropic turbulence in a periodic box at initial $Re_\lambda \approx 100$, and for the reconnection of two vortex tubes at $Re_\Gamma = 3500$. In the last case the flow was still laminar. For both cases it was found that the vortex method resolves the large- and medium scales in the flow well. In addition, the simulation of isotropic turbulence showed that the vortex method does not suffer from accumulation of energy in the tail of the energy spectrum, whereas the spectral method does. Furthermore, an under-resolved flow simulation of the colliding vortex tubes showed that the pseudo-spectral method generates spurious vortex structures, but with the vortex method the large scales are still adequately resolved and no spurious vorticity appears. In a previous work by Cocle, Bricteux, and Winckelmans (2009), a vortex method and a pseudo-spectral code were used to compare various multiscale sub-grid models in LES of homogeneous isotropic turbulence. They report little difference between the two methods in the obtained spectra when using the same sub-grid model.

In this study we focus on the accuracy of the vortex method at higher Reynolds numbers than in Cottet, Michaux, et al. (2002), and we study the effect of employing a higher order remeshing kernel in the remeshed vortex method. This work is organized as follows. First we describe the methods used in this study. Then we report on the simulation results for a Taylor-Green vortex at $Re_\Gamma = 1600$. Finally we move on to a flow case at $Re_\Gamma = 10\,000$ and describe the comparison between the vortex method results and the pseudo-spectral method results.

3.2 Governing equations and numerical method

3.2.1 Vortex method

The evolution of viscous incompressible flow is considered as described by the Navier-Stokes equations in Lagrangian vorticity form

$$\frac{\mathrm{D}\boldsymbol{\omega}}{\mathrm{D}t} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\Delta\boldsymbol{\omega}, \tag{3.1}$$

and

$$\Delta\boldsymbol{\Psi} = \nabla\times\mathbf{u} = -\boldsymbol{\omega}, \tag{3.2}$$

where $\boldsymbol{\Psi}$ is the vector streamfunction.

The equations are discretized using a remeshed vortex method (rVM). In the traditional vortex particle method, the vorticity field is approximated using particles

$$\boldsymbol{\omega}(\mathbf{x},t) \approx \sum_p \boldsymbol{\Gamma}_p(t)\zeta_\epsilon(\mathbf{x}-\mathbf{x}_p(t)), \tag{3.3}$$

where $\boldsymbol{\Gamma}_p(t)$ and $\mathbf{x}_p(t)$ denote the particle strength and particle position, respectively, of the p-th particle at time t. In our hybrid formulation of the vortex particle method, the kernel function ζ_ϵ is used for interpolation between the particles and the grid (see the next subsection). To compute the Fourier-transform of our quantities, we assume a Fourier interpolation on the grid rather than using equation 3.3.

Discretizing the Navier-Stokes equations with particles results in a set of ordinary differential equations (ODEs) for the particle strengths and the particle positions

$$\frac{\mathrm{d}\boldsymbol{\Gamma}_p}{\mathrm{d}t} = v_p(\boldsymbol{\omega}\cdot\nabla^h)\mathbf{u} + \nu\Delta^h\boldsymbol{\Gamma}_p, \tag{3.4}$$

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}_p,t). \tag{3.5}$$

Here $v_p = h^3$ are the particle volumes. These differential equations are integrated in time using either a low-storage third-order Runge-Kutta method (J. Williamson 1980) (RK3), or a fourth-order Runge-Kutta method (RK4). The discretized operator for the viscous term is evaluated with a centred fourth-order finite-difference scheme. The stretching term is rewritten in its transpose formulation $\omega_k \partial u_k / \partial x_i$, and is discretized with fourth-order finite differences. Every time step the particles are remeshed onto a uniform Cartesian grid to enforce that the particles always overlap. In this way the occurrence of spurious vortical structures is prevented and convergence of the method is ensured (Koumoutsakos 1997; Cottet and Koumoutsakos 2000). The velocities are computed from the vorticity by solving equation (3.2) in Fourier space on the grid. This ensures that the velocity field is spectrally divergence-free. To ensure a divergence-free vorticity field ($\nabla \cdot \omega = 0$), a solenoidal reprojection based on the Helmholtz decomposition of the vorticity field is done in spectral space every 5 to 10 time steps.

The time step varies during the computation according to the deformation in the flow. Specifically, with the deformation tensor defined as $S = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$, then the time step according to the deformation criterion is given by the one-norm of S

$$\delta t = \mathrm{LCFL} \left(\max_{1 \le j \le 3} \sum_{i=1}^{3} |S_{ij}| \right)^{-1}, \tag{3.6}$$

where LCFL is a user-controlled parameter. This measure bounds the deformation rate in the flow. A second criterion on the time step is given by the Fourier number and relates to the stability of the diffusion term discretized on the mesh. For high Reynolds number flows the magnitude of the time step is dominated by the deformation measure.

The vortex code is a parallel application implemented as a client of the Parallel Particle Mesh library (PPM, Sbalzarini et al. 2006).

3.2.2 Kernel function for vortex method

In our method, we use the particle kernel function $\zeta_\epsilon(\mathbf{x} - \mathbf{x}_p(t))$ for the interpolation between the particle and the mesh. We use $\epsilon = h$, where h is the spacing of the (uniform) grid. A commonly used kernel in particle methods is the M_4' kernel (Monaghan 1985; Koumoutsakos 1997; Winckelmans 2004), defined as

$$M_4'(x) = \begin{cases} 0 & \text{if } |x| > 2, \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 < |x| \le 2, \\ 1 - \frac{5}{2}x^2 + \frac{3}{2}|x|^3 & \text{if } |x| \le 1. \end{cases} \tag{3.7}$$

This kernel has a four-point support and has an error of $\mathcal{O}(h^3)$. In addition to the M_4' kernel, we will use a kernel derived in Bergdorf (2007), denoted as the M_6^* kernel. This kernel has a support of six points and has an error of $\mathcal{O}(h^4)$. It is defined by

$$M_6^*(x) = \begin{cases} 0 & \text{if } |x| > 3, \\ -\frac{1}{24}(|x|-2)(|x|-3)^3(5|x|-8) & \text{if } 2 < |x| \leq 3, \\ \frac{1}{24}(|x|-1)(|x|-2)(25|x|^3 - 114x^2 + 153|x| - 48) & \text{if } 1 < |x| \leq 2, \\ -\frac{1}{12}(|x|-1)(25x^4 - 38|x|^3 - 3x^2 + 12|x| + 12) & \text{if } |x| \leq 1. \end{cases} \tag{3.8}$$

Both kernels are plotted in figure 3.1. In three dimensions we use the tensorial product of the above interpolation kernels. From here on, we will denote the remeshed vortex method with the M_4' interpolation kernel as rVM-M_4', the remeshed vortex method with the M_6^* interpolation kernel as rVM-M_6^* and the pseudo-spectral method as PS.

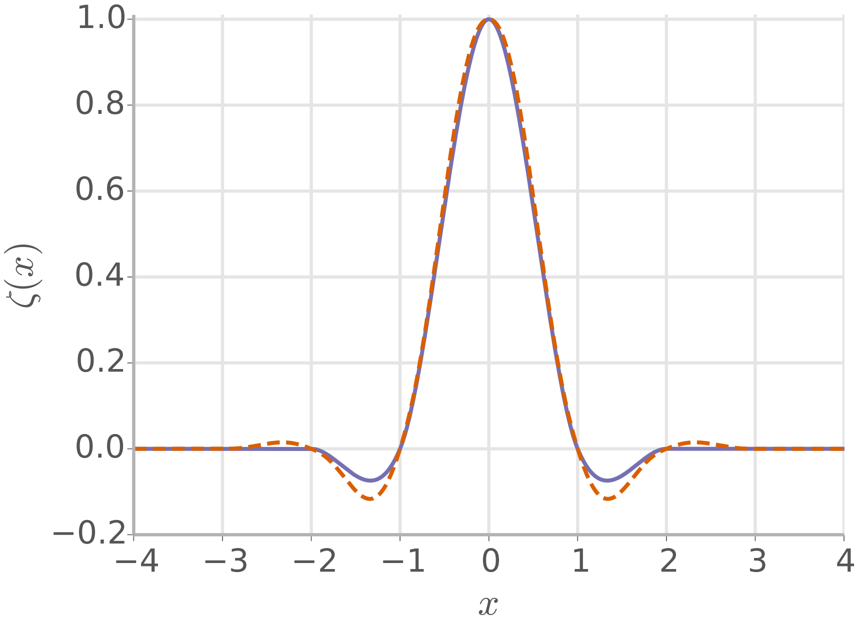


Figure 3.1: The four-point M_4' kernel (solid blue) and the six-point M_6^* kernel (dashed red) used for remeshing the particles onto the mesh in the rVM.

3.2.3 Pseudo-spectral method

We employ a standard pseudo-spectral method (PS) using a fourth-order Runge-Kutta method for time stepping. The Navier-Stokes equations for the vorticity are solved in the rotational formulation

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nabla \times (\mathbf{u} \times \boldsymbol{\omega}) + \nu \Delta \boldsymbol{\omega}, \tag{3.9}$$

together with the Poisson equation for the velocity given in equation (3.2). The rotational formulation ensures that the vorticity field remains divergence-free for the duration of the computation. De-aliasing is done with the smooth truncation formula described in Hou and Li (2007). For more details on the pseudo-spectral method, see Boyd (2000), Canuto et al. (2006), and Canuto et al. (2007). All pseudo-spectral simulations are performed with a constant CFL-number of 0.75. Like the vortex-particle method, our pseudo-spectral solver is implemented as a client of the PPM-library and exploits the scaling capacities of this library.

3.3 Taylor-Green vortex

The Taylor-Green vortex is a benchmark flow case that is often used to study the generation of small-scale vorticity and small-scale vortical structures from a smooth initial condition under the influence of vortex stretching (e.g. Brachet et al. 1983). The family of initial conditions, defined in a periodic cube with side length 2π, are described by the following equations

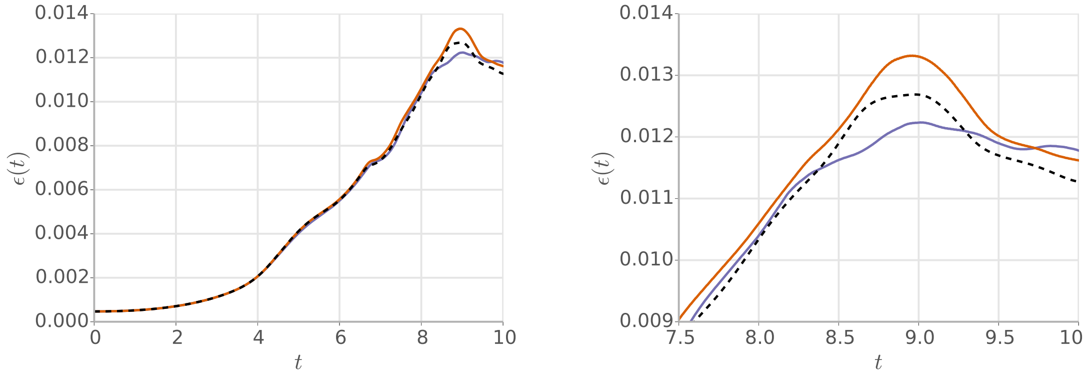
$$\begin{aligned} u_x(\mathbf{x}, t=0) &= \frac{2}{\sqrt{3}} \sin\left(\theta + \frac{2\pi}{3}\right) \sin(x)\cos(y)\cos(z), \\ u_y(\mathbf{x}, t=0) &= \frac{2}{\sqrt{3}} \sin\left(\theta - \frac{2\pi}{3}\right) \cos(x)\sin(y)\cos(z), \\ u_z(\mathbf{x}, t=0) &= \frac{2}{\sqrt{3}} \sin(\theta)\cos(x)\cos(y)\sin(z). \end{aligned} \tag{3.10}$$

Here θ is the free parameter, and it is set to $\theta = 0$ for all results in this study. The Reynolds number in this flow is defined as $Re_\Gamma = 1/\nu$ and in this work we set it equal to $Re_\Gamma = 1600$. The time integration for the vortex method is done with a third-order Runge-Kutta method with the time step determined by the minimum of the advection-based criterion, here $\mathrm{LCFL} = 0.125$, and the diffusion based criterion as determined by the Fourier number. For the pseudo-spectral method we employ a fourth-order Runge-Kutta time integration with $\mathrm{CFL} = 0.75$.

We performed the computation with the rVM-M_4', the rVM-M_6^* and the PS, using resolutions of 256^3, 512^3 and 768^3. The 256^3 resolution results are relevant as they shows how close the vortex method approximates the pseudo-spectral results for an under-resolved computation. At a resolution of 512^3 the pseudo-spectral method has converged. For each resolution we compare the vortex method results with the pseudo-spectral method results in the next subsections.

3.3.1 Comparison at 256^3 resolution

Figure 3.2 shows the evolution of the dissipation as a function of time. Up to $T \approx 6.5$, the rVM-M_6^* results are on top of the PS results, whereas the results from the rVM-M_4' slightly underpredict the dissipation. For larger times, the rVM-M_6^* results show a consistent larger dissipation than the PS results, although the shape of these two curves are qualitatively similar. In the close-up of the region $7.5 \leq T \leq 10.0$, it can clearly be seen that the rVM-M_6^* overpredicts the dissipation peak, whereas with the rVM-M_4' the dissipation peak is underpredicted with respect to the PS. Figure 3.6a shows the tail of the energy spectrum for all three methods at $T = 9.0$. The drop in energy at the highest wavenumbers in the pseudo-spectral method is due to the smoothing filter used for dealiasing, as mentioned above. We can see that in the high-end of the wavespectrum, the rVM-M_6^* slightly overpredicts the energy in the highest wavenumbers, whereas the rVM-M_4' underpredicts the energy.



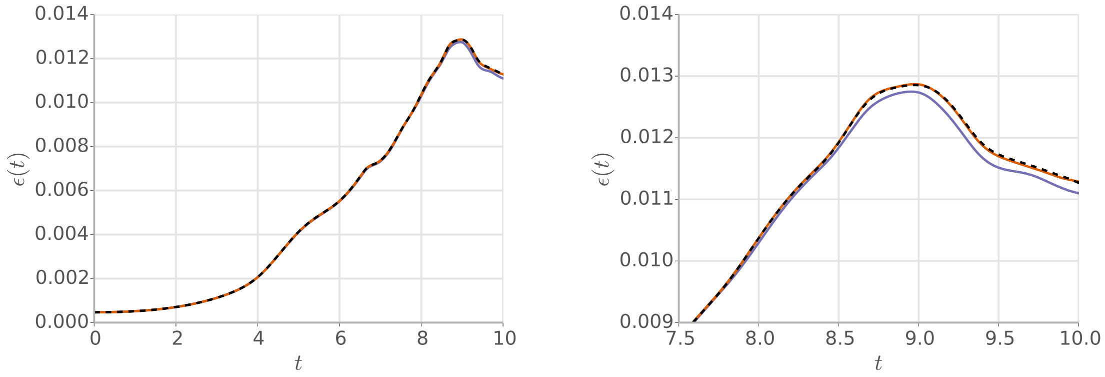
(a) overview

(b) close-up of region $7.5 \leq t \leq 10$

Figure 3.2: Energy dissipation of Taylor-Green vortex as a function of time, $Re_\Gamma = 1600$, the resolution is 256^3 for all methods. PS results (dashed black) compared with the rVM-M_4' results (blue) and with the rVM-M_6^* results (red).

3.3.2 Comparison at 512^3 resolution

The Taylor-Green flow has been computed for all methods with a resolution of 512^3. In figure 3.3 the evolution of the energy dissipation as a function of time is given for the rVM-M_4', for the rVM-M_6^* and for the PS. The rVM-M_4' at this higher resolution compares much better against the pseudo-spectral method, up to $T \approx 8.5$ there is a maximum difference between the two methods of 0.8%. Just like in the low-resolution case, the dissipation computed with the rVM-M_4' is consistently lower than the dissipation computed with the PS. Beyond $T \approx 8.5$ the deviation between the two methods increases as shown in the right pane of figure 3.3. Figure 3.4 shows the contours of maximum vorticity in the $x = 0$ plane for both the PS and the rVM-M_4', both at the same resolution of 512^3. The contourlines at $T = 8.0$ show that the flow computed by the vortex method approaches the pseudo-spectral flow, but visible discrepancies remain. One example is the overprediction of the vortex structure by the vortex method in the center bottom of figure 3.4a. At $T = 9.0$ the differences between the two methods have increased further.



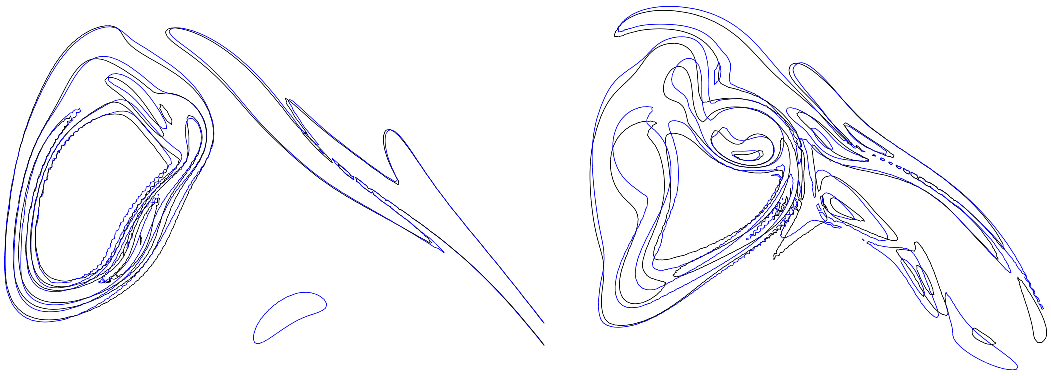
(a) overview

(b) close-up of region $7.5 \leq t \leq 10$

Figure 3.3: Energy dissipation of Taylor-Green vortex as a function of time, $Re_\Gamma = 1600$, the resolution is 512^3 for all methods. PS results (dashed black) compared with the rVM-M_4' results (blue) and with the rVM-M_6^* results (red).

The comparison of the results from the rVM-M_6^* with the PS results shows that the vortex method benefits greatly from the higher order kernel. The evolution of energy dissipation (figure 3.3) is virtually indistinguishable between the two methods, in the range $0 \leq T \leq 10$ the maximum difference between the M_6^* dissipation and the pseudo-spectral dissipation is 0.27%, occurring at time $T = 9.82$. Comparing the contours of vorticity magnitude (see figure 3.5) we see that the contour lines corresponding to higher vorticity values are indistinguishable. For the lower vorticity values some small differences persist (notice that the contour corresponding to the earlier mentioned vortex structure in the center bottom part of the plot at $T = 8.0$ persists for the M_6^* kernel computation), but overall the flow computed with the rVM-M_6^* agrees well with the PS results.

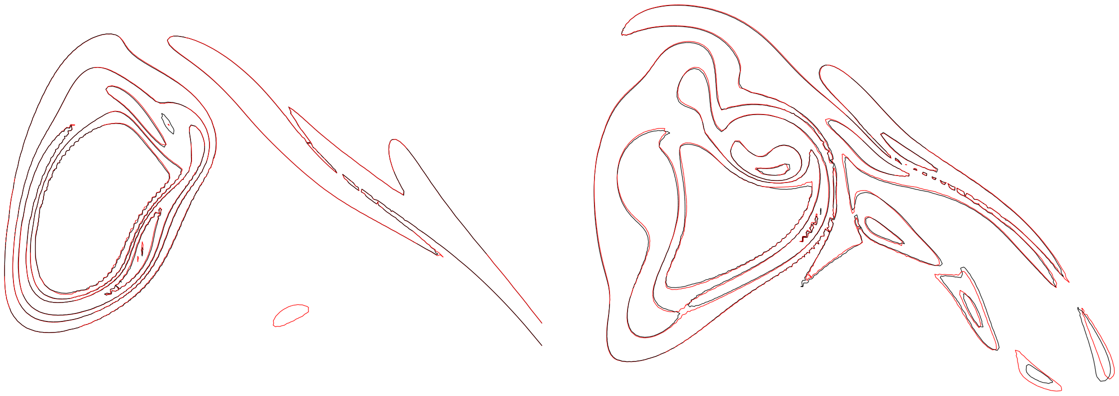
In figure 3.7a we show the highest wavenumbers of the energy spectrum at $T = 9.0$ for the methods at resolution 512^3. We can see that compared with the M_4' kernel, the vortex method results with the M_6^* kernel follow the pseudo-spectral results much closer in the



(a) $T = 8.0$

(b) $T = 9.0$

Figure 3.4: Contours of $|\boldsymbol{\omega}|$ in the $x = 0$ plane of the Taylor-Green vortex at $Re_\Gamma = 1600$. Shown are the PS results (black) and the rVM-M_4' results (blue), both with 512^3 resolution. The contours plotted are for $|\boldsymbol{\omega}|$ values of 1, 5, 10, 20 and 30.

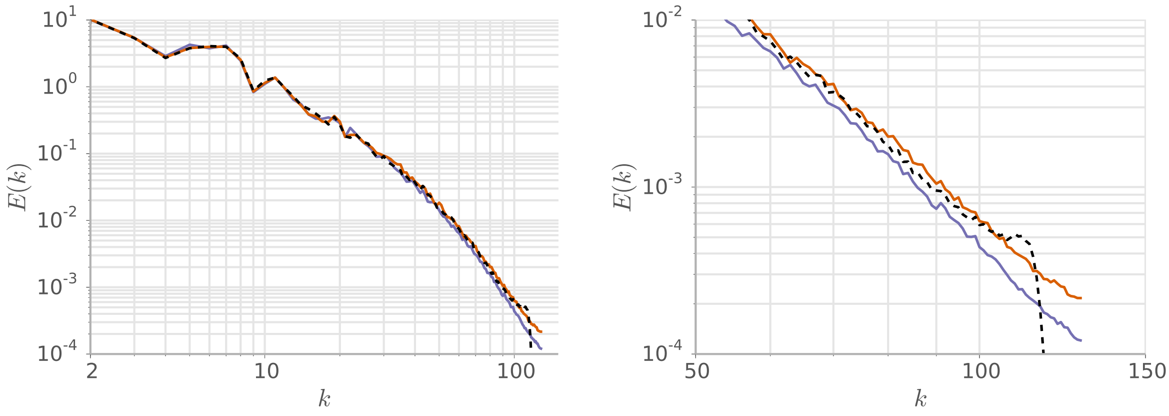


(a) $T = 8.0$

(b) $T = 9.0$

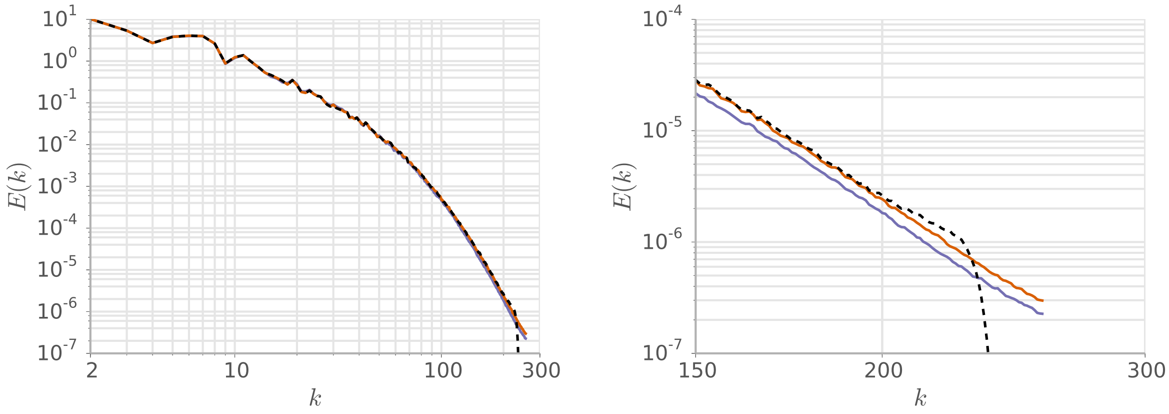
Figure 3.5: Contours of $|\boldsymbol{\omega}|$ in the $x = 0$ plane of the Taylor-Green vortex at $Re_\Gamma = 1600$. Shown are the PS results (black) and the rVM-M_6^* results (red), both with 512^3 resolution. The contours plotted are for $|\boldsymbol{\omega}|$ values of 1, 5, 10, 20 and 30.

range $100 \le k \le 200$. For wavenumbers $k > 200$ some differences start to appear. Since the pseudo-spectral method results are affected by the truncation (see the ‘hump’ in the energy spectrum just before it drops), we can not tell which method produces the more accurate energy spectrum in this regime.



(a) (b)

Figure 3.6: Comparison of the complete energy spectrum of the Taylor-Green vortex at $Re_\Gamma = 1600$, at $T = 9.0$ for 256^3 (left), and zoom in at the tail (right). Plotted are the PS results (dashed black), the rVM-M_4' results (blue) and the rVM-M_6^* results (red).



(a) (b)

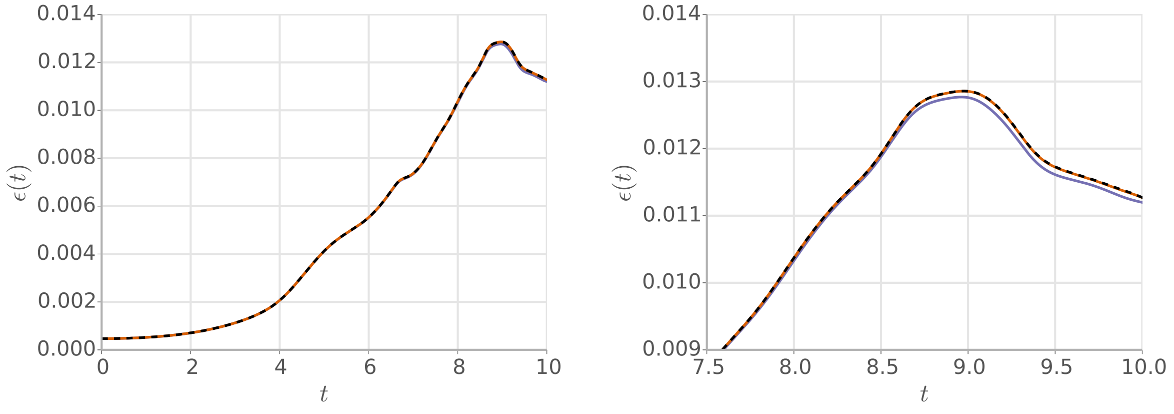
Figure 3.7: Comparison of the complete energy spectrum of the Taylor-Green vortex at $Re_\Gamma = 1600$, at $T = 9.0$ for 512^3 (left), and zoom in at the tail (right). Plotted are the PS results (dashed black), the rVM-M_4' results (blue) and the rVM-M_6^* results (red).

3.3.3 Comparison at 768^3 resolution

Figure 3.8 shows the dissipation evolution for the three methods at the highest resolution. The results for the vortex method with M_6^* kernel and the pseudo-spectral method overlap. The rVM-M_4', however, still slightly underpredicts the dissipation peak (see close-up). The contour lines of all three flows are plotted in figure 3.9 for times $T = 8.0$ and $T = 9.0$.

There is a minimal visual discrepancy between the rVM-M_6^* and the PS, only in the contour of the smallest vorticity some differences are visible. For the rVM-M_4' there is still a small difference present, which can only be explained by the lack of spatial accuracy of the interpolation kernel, even at a resolution of the background grid of 768^3.

The Taylor-Green simulations show that the rVM-M_6^* is able to capture a significant amount of flow scales and its accuracy approaches the accuracy of a spectral method for this specific flow-scenario. We will apply these results to a high Reynolds number simulation of a vortical flow in the next section.



(a) overview

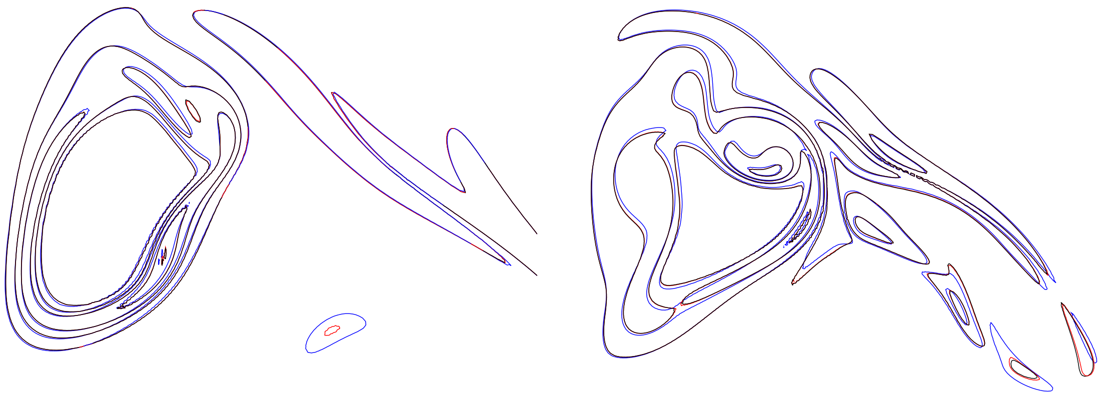
(b) close-up of region $7.5 \leq t \leq 10$

Figure 3.8: Energy dissipation of Taylor-Green vortex as a function of time, $Re_\Gamma = 1600$, the resolution is 768^3 for all methods. PS results (dashed black) compared with the rVM-M_4' results (blue) and with the rVM-M_6^* results (red).

3.4 Collision of two vortex tubes

We consider the flow evolving from two perturbed vortex tubes starting side by side in a periodic domain, for a Reynolds number based on the initial circulation of one vortex tubes of $Re_\Gamma = 10\,000$. This flow case exhibits many features which are common to turbulent flows such as vorticity topology change, a strong production of small-scale vorticity and its dissipation in the high-wavenumber range.

In a domain with size $[6\pi \times 4\pi \times 2\pi]$, the vortex tubes are initialized in a configuration commonly used in literature, e.g. Melander and Hussain (1988), Virk, Hussain, and Kerr (1995), and Carnevale et al. (1997). Specifically, we define the vorticity profile in the core



(a) $T = 8.0$

(b) $T = 9.0$

Figure 3.9: Contours of $|\boldsymbol{\omega}|$ in the $x = 0$ plane of the Taylor-Green vortex at $Re_\Gamma = 1600$. Shown are the PS results (black), the rVM-M_4' results (blue) and the rVM-M_6^* results (red), all with 768^3 resolution. The contours plotted are for $|\boldsymbol{\omega}|$ values of 1, 5, 10, 20 and 30.

as an axisymmetric function $\omega(r)$ as

$$\omega(r) = \begin{cases} \omega_0 \left[1 - \exp\left(-\frac{K}{r^*} \exp\left(\frac{1}{r^* - 1} \right) \right) \right] & \text{if } r^* < 1, \\ 0 & \text{else,} \end{cases} \tag{3.11}$$

where $r^* = r/r_{\text{cutoff}}$ and the factor K is defined as $K = 1/2 \exp(2) \log(2)$. Following Melander and Hussain (1988), we choose $r_{\text{cutoff}} = 0.666$ for our setup.

Along the z-direction the tubes are sinusoidally perturbed with an amplitude A at an inward inclination angle α. For the vortex tube in the negative half of the x-plane, the distance to the core is defined as

$$r(\mathbf{x})^2 = [x - (x_c + A\cos(\alpha)(1 + \cos(z)))]^2 + [y - (y_c + A\sin(\alpha)(1 + \cos(z)))]^2 , \tag{3.12}$$

where $x_c + 2A\cos(\alpha)$ and $y_c + 2A\sin(\alpha)$ are the x and y-coordinates of the center of the vortex tube in the $z = 0$ plane.

The vorticity field corresponding to this tube is then given by

$$\boldsymbol{\omega}(\mathbf{x}) = \omega(r)\,(-A\cos(\alpha)\sin(z)\hat{\mathbf{x}} - A\sin(\alpha)\sin(z)\hat{\mathbf{y}} + \hat{\mathbf{z}}) . \tag{3.13}$$

Here $\hat{\mathbf{x}}$,$\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ are the unit vectors in x, y and z directions, respectively.

The space between the unperturbed tubes in the $z = \pi$-plane is 1.732. We set $y_c = 0$ for both tubes, while $x_c = -0.866$ for the tube described above and $x_c = 0.866$ for the other tube. The perturbation amplitude is equal to $A = 0.2$ and the inclination angle of the tubes in the xy-plane, with respect to the y-axis, is equal to $\alpha = \pi/3$ for the tube in the negative half of the x-plane and $\alpha = 2\pi/3$ for the other. We evolve the flow with $\nu = 0.001$ and set $\omega_0 = 26.093$ to achieve a Reynolds number based on the initial circulation in the cross-section of one of the tubes equal to $Re_\Gamma = 10\,000$. Finally, after initialization, we perform a spectral solenoidal reprojection of the vorticity field.

Starting from these initial conditions, we evolve the flow three times: once with the PS, once with the rVM-M_4' and once with the rVM-M_6^*. In all cases the resolution of the grid is equal to [1280 × 960 × 640] points. For the PS computations we choose a constant CFL = 0.75. In the computation with the rVM-M_4' we use our RK3 time integration with LCFL = 0.0625. For the rVM-M_6^* computations we use a RK4 time integration with LCFL = 0.125. Each computation was done on 4096 cores on a Cray XT5 system at the Swiss National Supercomputer Center (CSCS).

3.4.1 Comparison between the pseudo-spectral method and the vortex method

Here we compare several diagnostics as well as flow visualizations at specific times between the vortex method results and the pseudo-spectral results. A measure of the accuracy of the computations at any point in time is provided by the effective viscosity of the flow. From integrating the evolution equation of the kinetic energy in the flow over the computational domain, it follows that (Winckelmans and Leonard 1993)

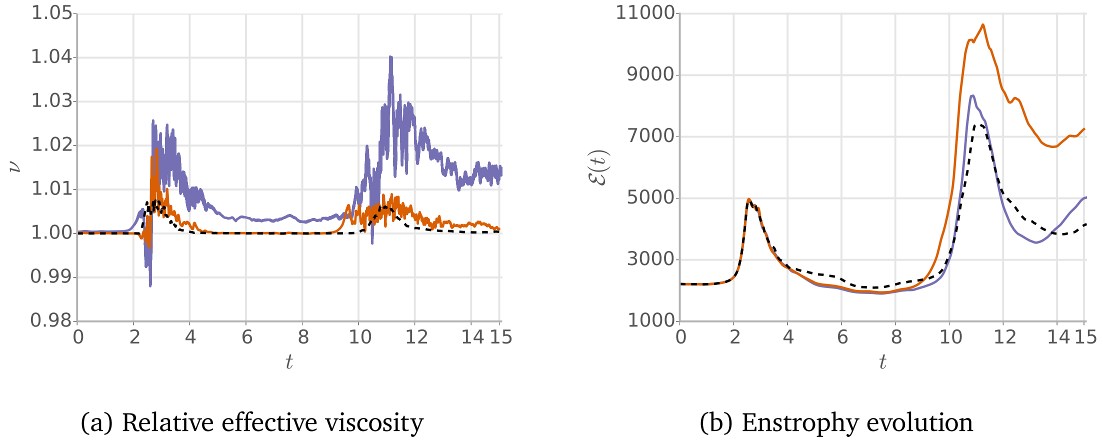
$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\nu\mathcal{E}, \tag{3.14}$$

where $\mathcal{E} = \int \boldsymbol{\omega} \cdot \boldsymbol{\omega} \, \mathrm{d}\mathbf{x}$ is the enstrophy and E is the kinetic energy. The effective viscosity can then be defined by

$$\nu_{\text{effective}} = -\frac{\mathrm{d}E}{\mathrm{d}t}/\mathcal{E}. \tag{3.15}$$

The energy and enstrophy are computed from the velocity and vorticity on the mesh, respectively. We compare the effective viscosity to the actual viscosity in the flow to measure the error.

The results for the computations are given in figure 3.10a, where we divided the effective viscosity by the physical viscosity $\nu = 0.001$ to study the relative differences. The maximum error in effective viscosity is about 4% and 2.5% for the rVM-M_4' and the rVM-M_6^*, respectively, and about 1% for the pseudo-spectral method. An interesting element of this plot is that the vortex method, with both the M_4' and the M_6^* kernels, gives a relative effective viscosity smaller than 1 during some parts of the flow evolution. This means that our method is less dissipative than it should be according to the physical viscosity of the flow in these parts of the computation, and that we are actually computing at a slightly larger effective Reynolds number than the physical Reynolds number. The pseudo-spectral method, on the other hand, has a relative effective viscosity which is always larger than 1.

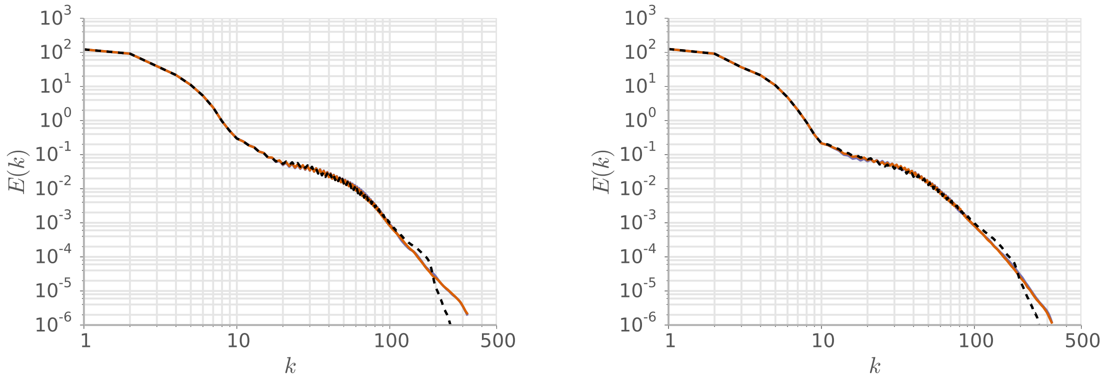


(a) Relative effective viscosity

(b) Enstrophy evolution

Figure 3.10: Comparison of the relative effective viscosity and the total enstrophy over time. PS (dashed black), rVM-M_4' (blue) and rVM-M_6^* (red).

Figure 3.10b gives the total enstrophy in the flow compared between the three methods. Up to and including the first reconnection ($T \leq 4.0$) the three methods show qualitatively similar behavior. This is confirmed when comparing snapshots of the flow during the first reconnection (figure 3.12). The flow visualization shows that the M_6^* kernel resolves more scales than the M_4' kernel, although still differences with respect to the pseudo-spectral method can be seen. At times greater than approximately 4.0, the vortex method results start to diverge from the PS results. In figure 3.13 we visualize the flow around $T = 11$, during the second reconnection. Since the second reconnection starts earlier for the rVM-M_6^*, in the left pane of figure 3.13 the reconnection is further developed for the rVM-M_6^* results compared with the other results. However, we should not forget that we are simulating a high Reynolds number flow initialized without noise. This means that small differences in the flow at one point in time can lead to very different flow evolutions later in time. Given the evolution of the effective viscosity (figure 3.10a) the rVM-M_6^* still approximates the Navier-Stokes equations with a smaller error than the rVM-M_4', so we can conclude that for later times the comparison between the different results should not be given too much importance.



(a) $T = 2.654$

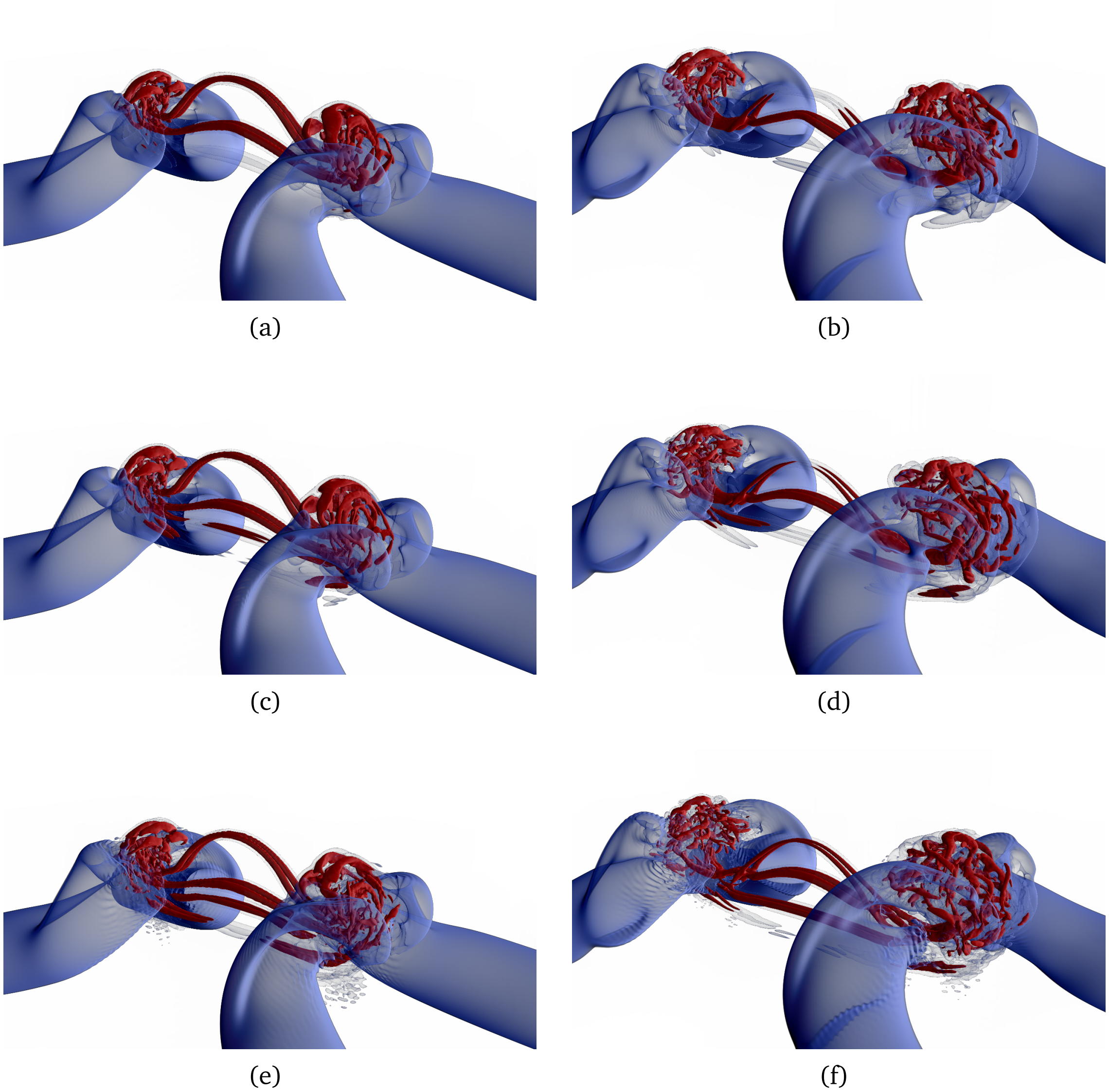
(b) $T = 2.850$

Figure 3.11: Energy spectrum at the times corresponding to figure 3.12. PS (dashed black), rVM-M_4' (blue) and rVM-M_6^* (red).

3.4.2 Timings

For the particular case of the colliding vortex tubes, we compare the timings between the pseudo-spectral method and the vortex method. As mentioned before, the implementation of the FFTs used in our pseudo-spectral method is the same as the FFTs used in the vortex method. Table 3.1 presents the results. The additional cost of the vortex method with RK4 algorithm over the pseudo-spectral method is because of the particle operation, predominantly particle mapping and particle interpolation. The difference is small, however, because the pseudo-spectral method needs to FFT more fields per substep of the algorithm. Each substep, the vortex method forward-transforms the vorticity and backward-transforms the velocity, but the pseudo-spectral method needs to backward-transform both the vorticity and the velocity, and then forward-transforms the term $\mathbf{u} \times \boldsymbol{\omega}$.

Because of the large time steps allowed with the vortex method the time to solution is smallest for the vortex method with M_6^* kernel, albeit at the cost of a larger error. We note that the timings presented here are very dependent on the flow case and computational settings. In fact, if we would double the resolution of this computation, the number of time steps required for the pseudo-spectral method increases by a factor of two due to the CFL criterion; for the vortex method, assuming that the deformation in the flow does not change much, we could use the same time step and therefore the number of time steps remains unchanged. Furthermore, we know from the scaling properties of our code (Chatelain, Curioni, et al. 2008) that the Fast Fourier Transforms (FFTs) are the main limitation in the scalability. Since the pseudo-spectral method performs more FFTs


(a)

(b)

(c)

(d)

(e)

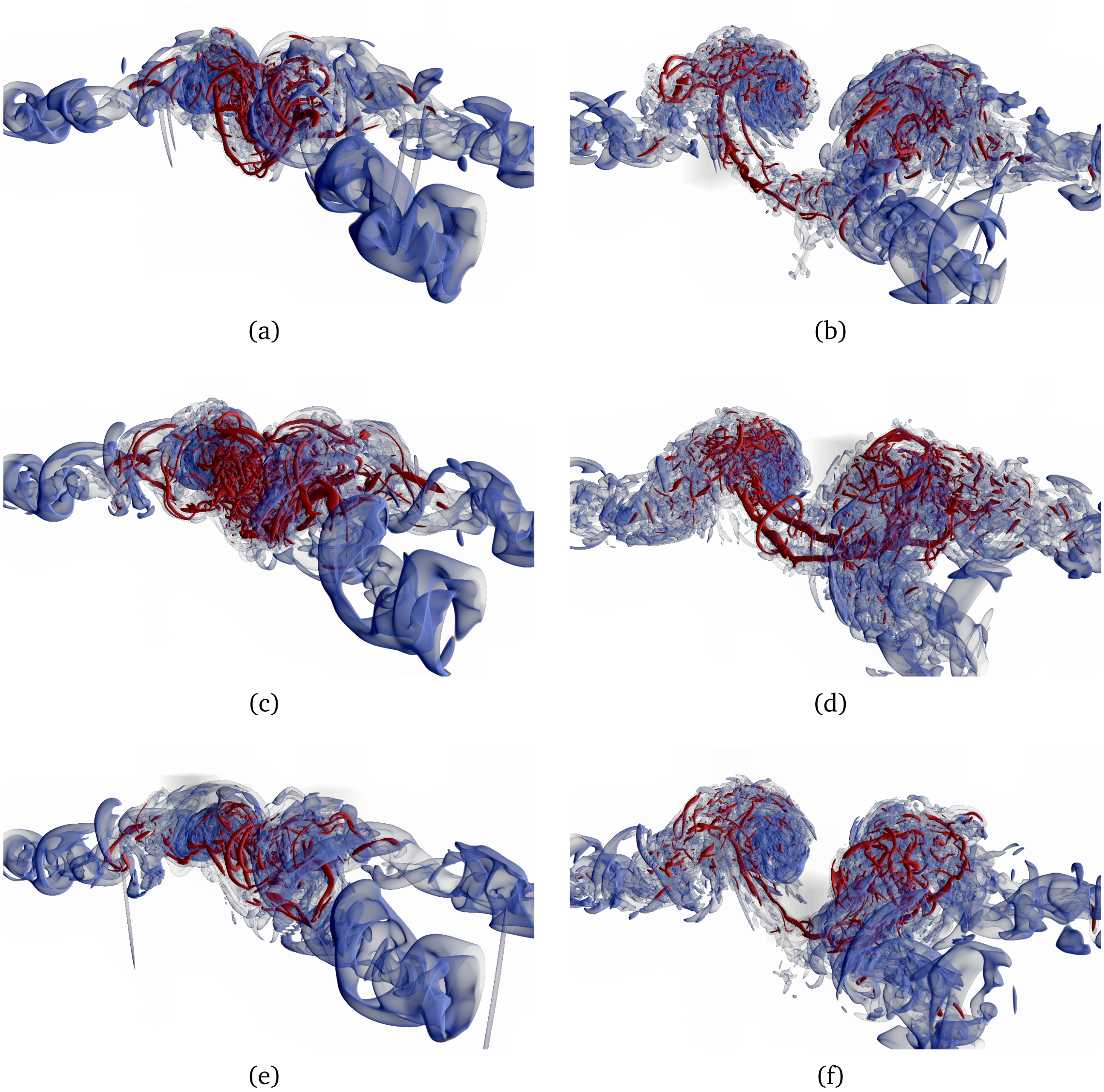
(f)

Figure 3.12: Visualization of the flow at time $T = 2.65$ (a),(c),(e) and at time $T = 2.85$ (b),(d),(f). Results predicted with rVM-M_4' (a),(b), with rVM-M_6^* (c),(d) and with PS (e),(f). The figure is a volumerendering of the λ_2-field (Jeong and Hussain 1995).

per time step than the vortex method, we expect a better scaling behavior for the vortex method at a larger number of cores.

3.5 Conclusions

We have compared the remeshed vortex method with a pseudo-spectral method in simulations of a Taylor-Green vortex at $Re_\Gamma = 1600$ and two colliding vortex tubes at $Re_\Gamma = 10\,000$.



(a) (b)

(c) (d)

(e) (f)

Figure 3.13: Visualization of the flow at time $T = 10.5$ (a),(c),(e) and at time $T = 11.5$ (b),(d),(f).Results predicted with rVM-M_4' (a),(b), with rVM-M_6^* (c),(d) and with PS (e),(f). The figure is a volumerendering of the λ_2-field (Jeong and Hussain 1995).

For both studies, the vortex method resolves the flow qualitatively well when the third-order M_4' kernel is used for interpolation between the particles and the mesh. When the kernel is replaced by the higher-order M_6^* kernel, the results of the vortex method match the results computed by the pseudo-spectral method also quantitatively in all but the smallest length scales.

For the high Reynolds flow case, we have established the remeshed vortex method as a fast and accurate alternative to the pseudo-spectral method.

method	time per time step (s)	number of time steps	time to solution (hrs)	max error in $\nu_{\text{effective}}$
rVM-M_4', RK3, LCFL $= 0.0625$	10.6	10034	29.5	4.0%
rVM-M_6^*, RK4, LCFL $= 0.125$	18.3	5398	27.4	2.5%
PS, RK4, CFL $= 0.75$	14.8	7185	29.5	1.0%

Table 3.1: Comparison of timings between vortex method and pseudo-spectral method for the collision of two vortex tubes at $Re_\Gamma = 10\,000$ on 4096 cores. We report the number of time steps to reach $T = 10$.

Acknowledgments This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s70. We wish to acknowledge many helpful discussions with Yue Yang (Caltech) during the course of this work. Figures 3.12 and 3.13 were rendered using the VolumePerception software (Rossinelli 2011).

CHAPTER 4

Reconnection of vortex tubes

In the previous chapter we established the rVM as a viable alternative to pseudo-spectral methods for the simulation of high Reynolds number vortical flows. The second test case of the comparison was the reconnection of two anti-parallel vortex tubes at circulation-based Reynolds number of $Re_\Gamma = 10\,000$.

In this chapter we consider that flow case from a vortex dynamics point of view. Previous detailed studies have only analyzed such vortex reconnections up to $Re_\Gamma = 3500$ and until just after the tubes first connect. Here we show our higher Reynolds number results up to late times and identify the dominant processes involved.

This chapter is based on van Rees, Hussain, and Koumoutsakos (2012).

4.1 Introduction

The reconnection of vortex tubes is an archetypal problem of fluid dynamics. Repeated vortex reconnections have been postulated as a driving mechanism for turbulence and aerodynamic noise generation (Hussain 1983) and the destruction of aircraft wakes (Crow 1970; Scorer and Davenport 1970). Studies of vortex reconnection have been central to the question of singularity formation of the Euler and Navier-Stokes equations (Pumir and Siggia 1987; Kerr 1993; Hou and Li 2007; Hou and Li 2008; Hou and Lei 2009).

Experimental studies of vortex reconnection include the pioneering works of (Fohl and Turner 1975; Oshima and Asaka 1975; Schatzle 1987; Oshima and Izutsu 1988) for the collision of two vortex rings. A number of simulations, predominantly based on spectral methods, have described the different phases of the reconnection process for Reynolds numbers in the range 1000–3500. These studies include reconnections of vortex rings (Ashurst and Meiron 1987; Kida, Takaoka, and Hussain 1989; Aref and Zawadzki 1991; Kida, Takaoka, and Hussain 1991), anti-parallel tubes (Pumir and Kerr 1987; Melander and Hussain 1988; Kerr and Hussain 1989; Shelley, Meiron, and Orszag 1993), anti-parallel tubes with an orthogonal offset (Melander and Zabusky 1988; Zabusky and Melander 1989; Zabusky, Boratav, et al. 1991; Boratav, Pelz, and Zabusky 1992) as well as other configurations leading to interesting topological dynamics (Kida and Takaoka 1988; Aref and Zawadzki 1991; Chatelain, Kivotides, and Leonard 2003). Head-on collision of two vortex rings has been studied experimentally and numerically (Oshima 1978; Kambe and Minota 1983; Stanaway, Shariff, and Hussain 1988; Lim and Nickels 1992; Chu et al. 1995). A theoretical model of vortex reconnection was proposed by Saffman (1990), and has been critically evaluated against numerical simulations in Boratav, Pelz, and Zabusky (1992) for the case of two orthogonal vortex tubes, and in Shelley, Meiron, and Orszag (1993) for the case of two anti-parallel vortex tubes. A review of several of these works has been presented in Kida (1994).

Recently Hussain and Duraisamy (2011) presented results of vortex reconnection up to $Re_\Gamma = 9000$. This work described the early time behavior of the vortices, limited to the first collision of the vortex tubes. In the current work we extend these simulations not only by studying a higher $Re_\Gamma = 10\,000$ but also by performing these simulations for longer times revealing for the first time the dynamics of the vortices after their first collision. We quantify the governing mechanisms and identify an important role of the axial flow (helicity) imparted to the vortex structures after their primary collision that governs their subsequent evolution. Furthermore in order to quantify the effects of axial flow in vortex reconnection we study the vortex reconnection process with an initially imposed axial flow. We observe a $-7/3$ slope in the energy spectrum in all cases where an axial flow is present during vortex collisions.

This work is organized as follows: in section 4.2 we detail the computational setup of our simulations. Section 4.3 explains the flow evolution without initial axial flow organized around the different phases in the flow. Section 4.4 presents the effects of initial axial flow on vortex reconnection. We present our conclusions in section 4.5.

4.2 Computational methods

Simulations are performed in the velocity ($\mathbf{u}$) – vorticity ($\boldsymbol{\omega} \equiv \nabla \times \mathbf{u}$) form of the incompressible Navier-Stokes equations

$$\nabla \cdot \mathbf{u} = 0, \tag{4.1}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \Delta \boldsymbol{\omega}, \tag{4.2}$$

where ν is the kinematic viscosity. These equations are discretized using a remeshed vortex method (Koumoutsakos 1997; Koumoutsakos 2005) as employed for massively parallel computer architectures (Sbalzarini et al. 2006; Bergdorf, Koumoutsakos, and Leonard 2007; Chatelain, Curioni, et al. 2008; van Rees, Leonard, et al. 2011).

We simulate the flow in a periodic domain with dimensions $[3\pi \times 4\pi \times 2\pi]$ centered at $(0,0,0)$ and an underlying mesh of $[1280 \times 960 \times 640]$ computational elements. The vorticity field is initialized to an axisymmetric profile (Melander and Hussain 1988)

$$\omega(r) = \begin{cases} \omega_0 \left[1 - \exp\left(-\frac{K}{r^*} \exp\left(\frac{1}{r^* - 1} \right) \right) \right] & \text{if } r^* < 1, \\ 0 & \text{else,} \end{cases} \tag{4.3}$$

where $r^* = r/r_{\mathrm{cutoff}}$ with $r_{\mathrm{cutoff}} = 0.666$, and $K = 1/2 \exp(2) \log(2)$. Along the z-direction the tubes are sinusoidally perturbed with an amplitude A and an inward inclination angle α to force their collision by self-induction. For the vortex tube in the negative half of the x-plane, the core location is

$$r(\mathbf{x})^2 = [x - (x_c + A\cos(\alpha)(1 + \cos(z)))]^2 + [y - (y_c + A\sin(\alpha)(1 + \cos(z)))]^2, \tag{4.4}$$

where $x_c + 2A\cos(\alpha)$ and $y_c + 2A\sin(\alpha)$ are the x and y-coordinates of the center of the vortex tube in the $z = 0$ plane. The vorticity field of this tube is given by (Virk, Hussain, and Kerr 1995; Carnevale et al. 1997)

$$\boldsymbol{\omega}(\mathbf{x}) = \omega(r) \left(-A\cos(\alpha)\sin(z)\mathbf{e}_{\mathrm{x}} - A\sin(\alpha)\sin(z)\mathbf{e}_{\mathrm{y}} + \mathbf{e}_{\mathrm{z}} \right). \tag{4.5}$$

Here $\mathbf{e}_{\mathrm{x}}$, $\mathbf{e}_{\mathrm{y}}$ and $\mathbf{e}_{\mathrm{z}}$ are the unit vectors in x, y and z directions, respectively.

The space between the unperturbed tubes in the $z = \pi$-plane is 1.732 and we set $A = 0.2$, $\alpha = \pi/3$ for the tube in the negative half of the x-plane and $\alpha = 2\pi/3$ for the other. Time is non-dimensionalized as $t = t^* \omega_0 / 20$, where ω_0 is the peak vorticity at $t^* = 0$. The $Re_\Gamma \equiv \Gamma/\nu = 10\,000$ is based on the initial circulation in each unperturbed tube. The initial vorticity field is rendered divergence free by a solenoidal reprojection.

The axial flow imposed initially on the tubes (see Section 4.4) is based on an axisymmetric velocity profile (similar to the vorticity profile)

$$u_z(r) = \begin{cases} u_0\left[1-\exp\left(-\frac{K}{r^*}\exp\left(\frac{1}{r^*-1}\right)\right)\right] & \text{if } r^* < 1, \\ 0 & \text{else,} \end{cases} \tag{4.6}$$

where $r^* = r/r_{\text{cutoff}}$ with $r_{\text{cutoff}} = 0.666$. The corresponding azimuthal vorticity distribution is

$$\omega_\theta(r) = \begin{cases} -\frac{u_0}{0.666}\left[\frac{K(r^{*2}-r^*+1)}{(r^*-1)^2 r^{*2}}\exp\left(\frac{1}{r^*-1}-\frac{K}{r^*}\exp\left(\frac{1}{r^*-1}\right)\right)\right] & \text{if } r^* < 1, \\ 0 & \text{else.} \end{cases} \tag{4.7}$$

The vorticity field obtained by adding this azimuthal vorticity component to the axial vorticity component is perturbed according to equation (4.4).

The accuracy of the simulations is quantified by the effective viscosity, defined as the ratio of the time decay of total energy and the total enstrophy (Winckelmans and Leonard 1993), which does not exceed 2% in all simulations. The results of the remeshed vortex methods for the reconnection without axial flow agree well with those of pseudo-spectral simulations. The reader is referred to chapter 3 for a comparison of pseudo-spectral and remeshed vortex methods.

4.3 Results without initial axial flow

The collision of the vortex tubes entails the distinct stages of: (a) the first vortex tube collision followed by (b) a quiescent evolution of the resulting elliptical vortex rings during which an axial flow is imparted to their core by the coiling of the vortex lines, and (c) the ensuing collision of the elliptical vortex rings. Visualizations of the vorticity field for all stages are shown in figure 4.1.

4.3.1 First collision

The vortex tubes approach and collide starting at $t = 2.5$. The tubes are flattened in the collision area forming two anti-parallel vortex sheets. Viscous cross-diffusion annihilates the circulation at the contact point and circulation transfers from the symmetry plane to the dividing plane (figure 4.2, left), leading to the formation of elliptical vortex rings. Similar to simulations at $Re_\Gamma = 1000$ (Melander and Hussain 1988), the connected transverse vortex lines pile up to form the transverse vorticity structures (*bridges*), and remnant

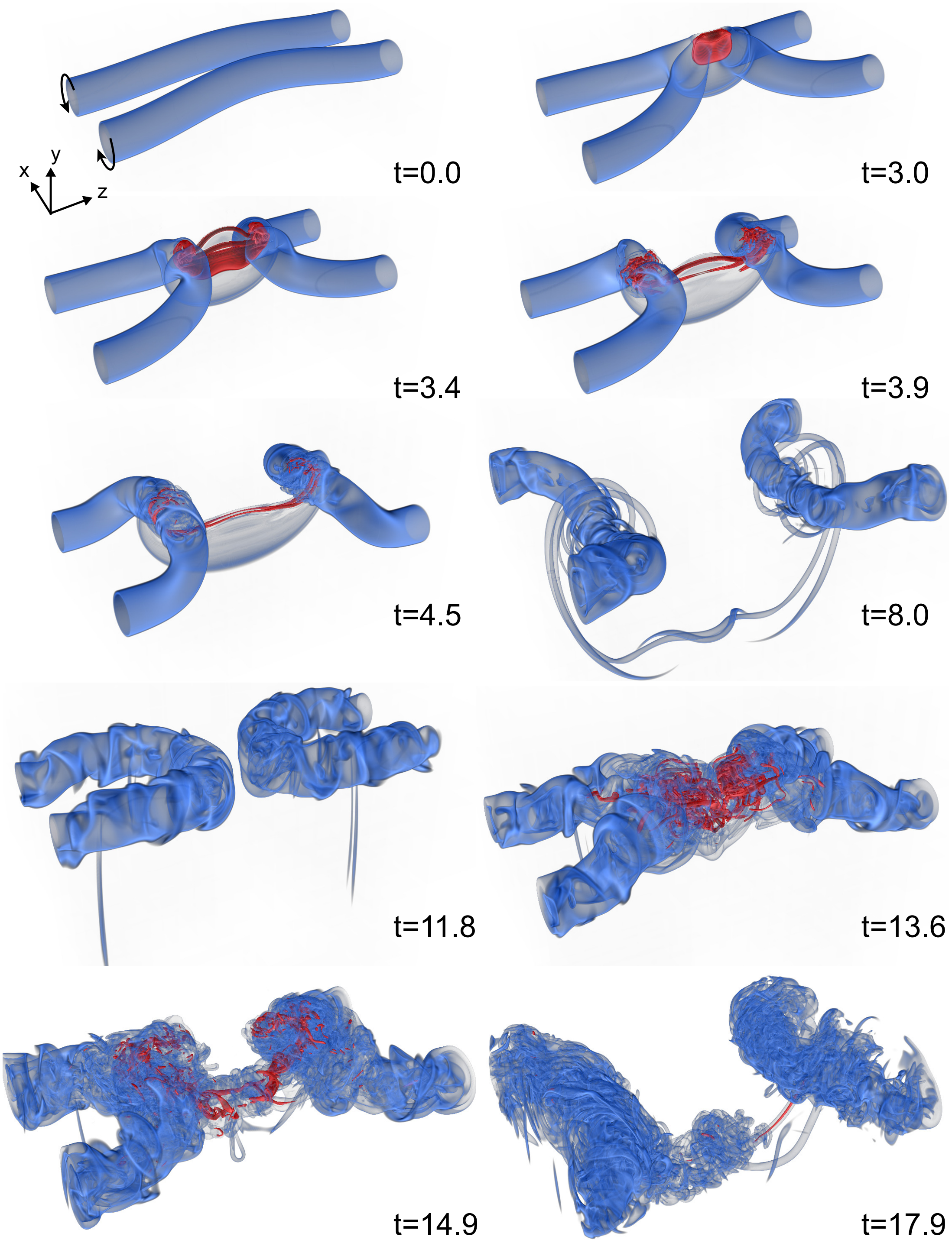


Figure 4.1: Volume rendering of vorticity magnitude at specified times. Blue and red are centered around values of $0.4\omega_0$ and $3.75\omega_0$, respectively.

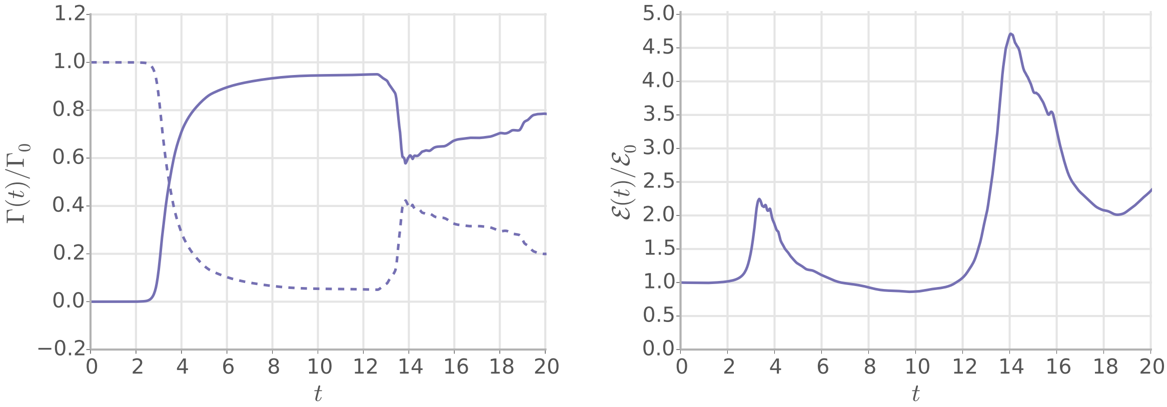


Figure 4.2: Left: Circulation in half of the planes defined by $x = 0$ (dividing plane, solid line) and $z = 0$ (symmetry plane, dashed line) as a function of time, normalized by the circulation in the $z = 0$ plane at $t = 0$. Right: Evolution of enstrophy over time, normalized by its value at $t = 0$.

circulation is organized into two parallel elongated structures (*threads*) consisting of vortex lines that did not undergo reconnection. As opposed to the lower Reynolds number case, (Melander and Hussain 1988), here the threads roll-up around the bridges (figure 4.5a) and acquire high vorticity due to their stretching by the bridges. Simultaneously, under mutual induction of adjacent bridges, the ends of the newly created vortex rings move up and away from each other. During this first phase of the reconnection process, the enstrophy rises to more than twice its initial value (figure 4.2, right).

We report for the first time a −5/3 slope in the energy spectrum of the flow (figure 4.3, left). While at $t = 0$ most energy is concentrated at the smallest wavenumbers, corresponding to the length scales of the tubes and the core diameters, at $t = 3.4$ we see a large increase in the small scale energy and the spectrum exhibits distinct power-law and viscous regimes. The power-law regime with −5/3 slope is manifested as soon as the tubes first touch at $t = 2.5$. At $t = 3.1$ the −5/3 slope is fully established in slightly less than a decade of intermediate wavenumbers and remains at these levels until $t = 3.4$. At later times the energy in the high wavenumbers gradually decays by dissipation and we can no longer identify distinct power-law and viscous regimes.

With the first collision an axial flow is imparted on the core of the newly formed rings. This flow is directed away from the bridges, as visualized in figure 4.4 by the helicity density in a moving reference frame $\boldsymbol{\omega} \cdot (\mathbf{u} - \mathbf{U})$, where, following Kida, Takaoka, and

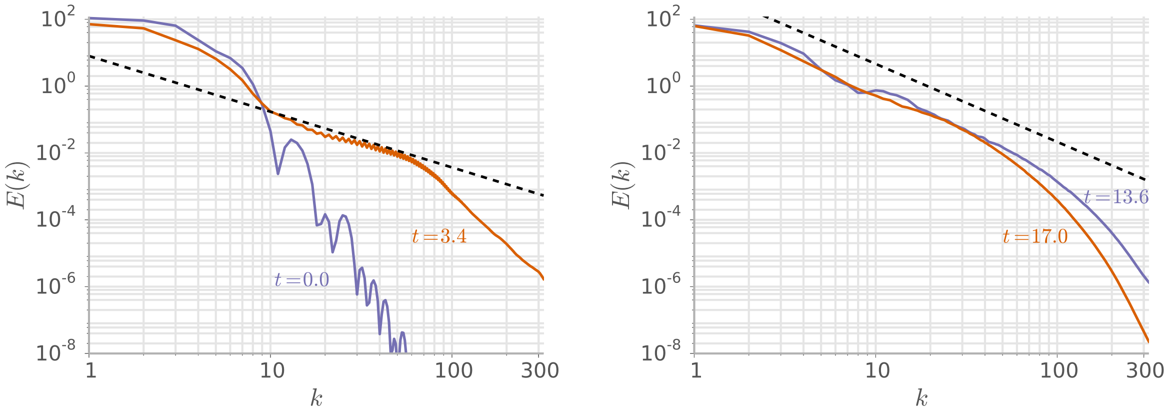


Figure 4.3: Three-dimensional energy spectrum of the case without initial axial flow. Left: $t = 0$ and $t = 3.4$, right: $t = 13.6$ and $t = 17.0$. The straight dashed lines correspond to a $-5/3$ slope in the left plot, a $-7/3$ slope in the right plot.

Hussain (1991), **U** is given by

$$\mathbf{U} = \left(0, \frac{\displaystyle\int (\boldsymbol{\omega} \cdot \mathbf{u}) \omega_2 \, \mathrm{d}V}{\displaystyle\int (\omega_2)^2 \, \mathrm{d}V}, 0 \right)^T . \tag{4.8}$$

In order to elucidate the establishment of the axial flow we visualize the evolution of the vorticity field by instantaneous vortex lines in figure 4.5. The lines in the primary reconnection region are rolled up by the swirl of the bridges. The intense vorticity of the threads induces in turn axial flow within the bridges away from the initial plane of symmetry (Melander and Hussain 1988) (figure 4.4).

4.3.2 Quiescent period

The quiescent period between $t = 4.0$ and $t = 12.0$ involves the evolution of the elliptical vortex rings (figure 4.6). At first, the rings have their major axes aligned with the z-direction. The ends of the threads, remnants of the first collision, wrap around the bridges. In this phase, the vorticity magnitude in these threads is much lower than in the bridges. Helical waves on the rings can be observed: first single waves traveling away from the connection region, and then several co-existing helical waves of opposite directionality. The elliptic vortex rings undergo an oscillation, switching axes, such that at $t \approx 12$ their major axes are again aligned with the z-direction.

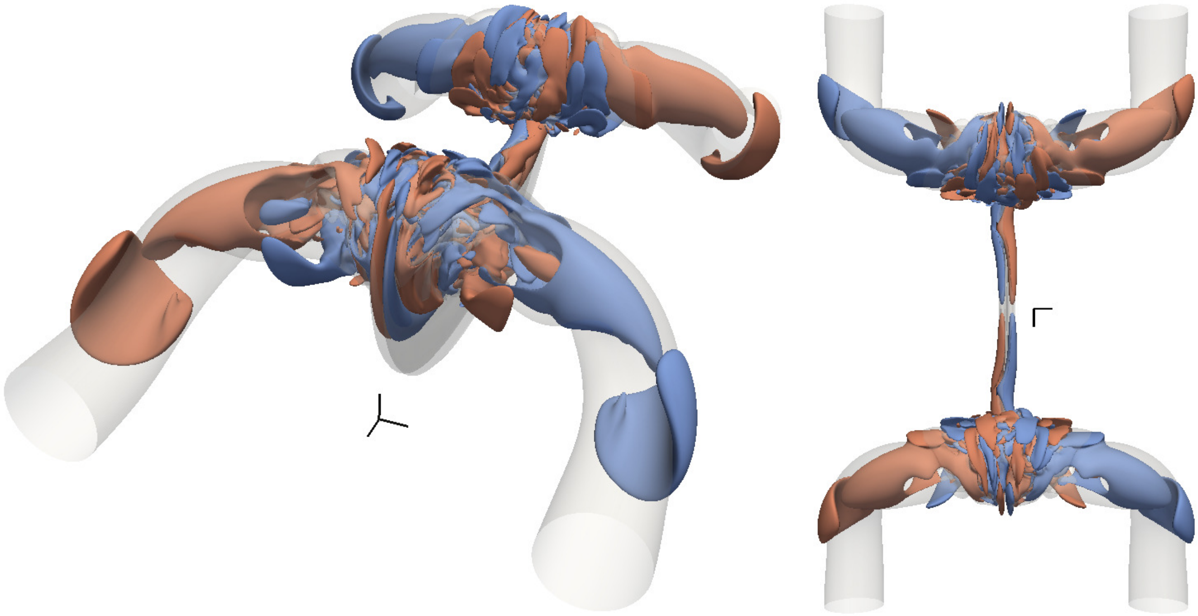


Figure 4.4: Two isosurfaces of $\boldsymbol{\omega} \cdot (\mathbf{u} - \mathbf{U})$ at ±2% of its maximum value at $t = 4.5$. Perspective view (left) and top view (right) with negative (light gray, blue online) and positive (dark gray, red online) isosurfaces and transparent isosurface of $|\boldsymbol{\omega}| = 0.5\omega_0$ (gray).

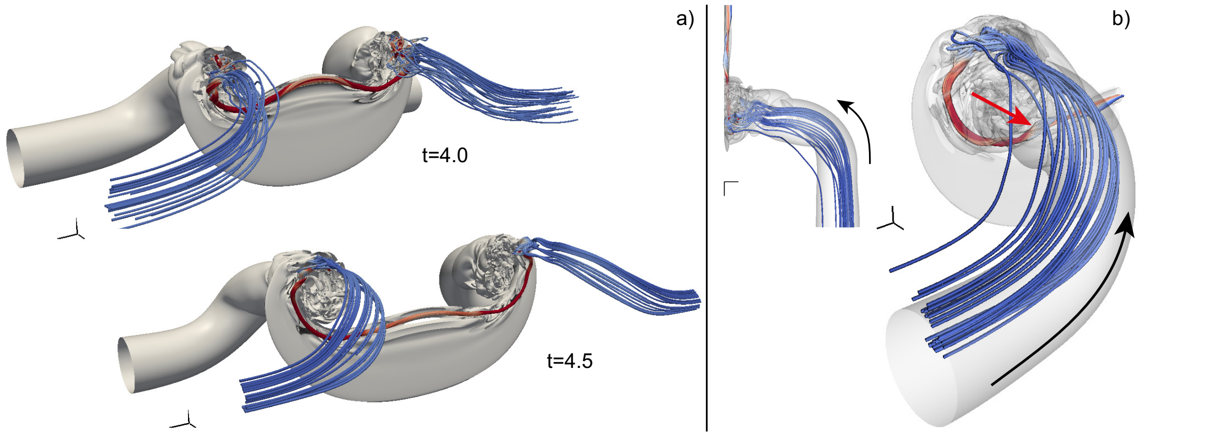


Figure 4.5: (a) Vortex lines and an isosurface of $|\boldsymbol{\omega}| = 0.5\,\omega_0$ (solid gray). The vortex lines, colored by $|\boldsymbol{\omega}|$, are seeded from regions where $|\boldsymbol{\omega}| > 7.0\,\omega_0$ and are drawn only for $\boldsymbol{\omega} > 0.25\,\omega_0$. (b) Directions of vorticity (black arrows) and axial flow (gray arrow, red online) for the $x > 0$ and $z > 0$ quarter of the domain at $t = 4.5$.

We note that the energy dissipation after the collision follows a decay law of the form

$$\frac{\partial E}{\partial t} \propto -t^{-(p+1)}, \tag{4.9}$$

where p is in the range $p = 1.2$–1.4 (figure 4.7), similar to the energy decay rate found in

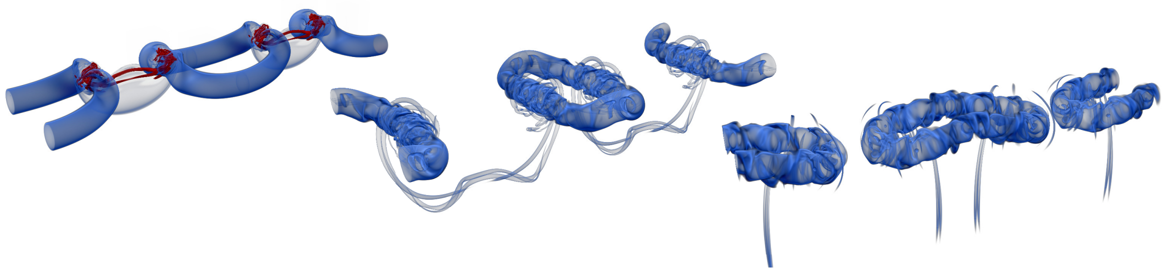


Figure 4.6: Volume renderings of vorticity magnitude plotted for $t = 3.0$ (left), $t = 5.4$ (middle) and $t = 8.4$ (right). Two periods in z-direction are shown, the colormap is as in figure 4.1.

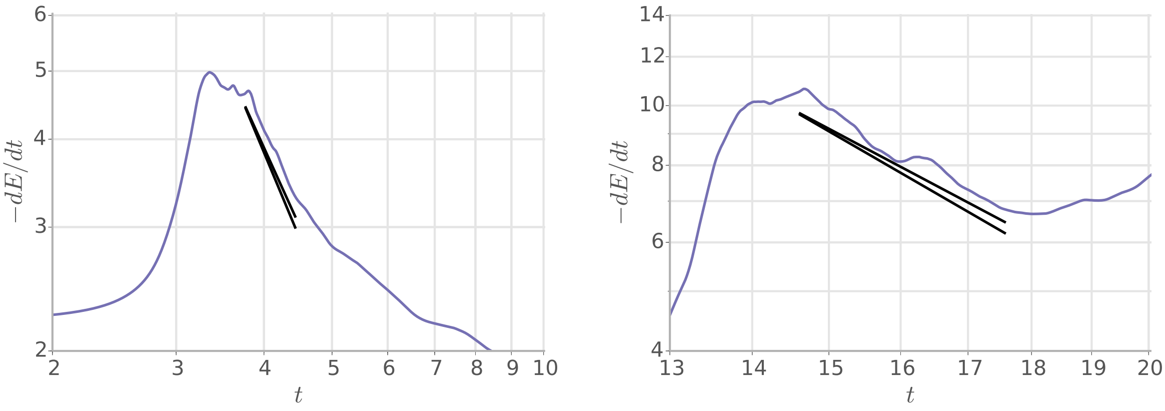


Figure 4.7: Energy decay after the first (left) and second (right) reconnection, on a log-log scale. The two solid lines in each panel denote slopes of -2.2 and -2.4, respectively.

fully developed turbulence, where $1.0 \lesssim p \lesssim 1.4$ (Kolmogorov (1991) and discussion in Lesieur (2008)). This result is consistent with observations made in Kida, Takaoka, and Hussain (1991) despite the much lower $Re_\Gamma = 1000$ of that study.

The axial flow in the core of the rings, originally directed away from the connection region (figure 4.4), changes its direction as the rings separate. A second reversal of the axial flow is observed starting at $t = 6.5$, such that it is again directed away from the connection region. For simplicity we represent the quarter ring in the $x > 0$ and $z > 0$ part of the domain by a straight vortex tube (figure 4.8), with the left end corresponding to the location of the bridge and the right end corresponding to the undisturbed end of the ring. Without any axial vorticity variation the vortex lines are straight (uncoiled) and no axial flow is imparted on the cores. After the collision, however, the vorticity in the bridges is stronger than the vorticity farther away, leading to a right-handed polarization of the

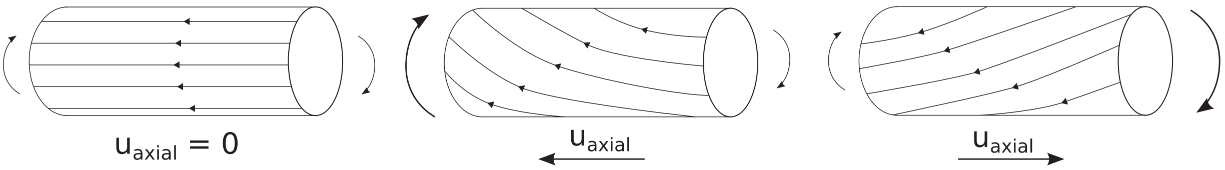


Figure 4.8: The effect of an axial vorticity gradient in the core of a vortex tube. Reference configuration (no axial flow) with constant vorticity and parallel vortex lines (left). A configuration with stronger vorticity on the left-side gives rise to a right-handed polarization in the vortex lines and an axial flow from right to left (middle). A stronger vorticity on the right-side gives rise to a left-handed polarization in the vortex lines and an axial flow from left to right (right).

vortex lines and hence an axial flow directed from right to left (middle of figure 4.8), i.e. towards the bridges. Subsequently, as the rings oscillate due to axes switching, the vorticity magnitude decreases in the bridges and increases at the undisturbed ends, consistent with elliptical vortex ring dynamics (Dhanak and Bernardinis 1981). The situation then resembles the right panel of figure 4.8: a left polarization of the vortex lines is established, imparting an axial flow directed away from the bridges. Note that in an experimental study on vortex rings (Naitoh et al. 2002), an axial flow inside the ring was also found, and attributed to the presence of an axial vorticity gradient.

In order to further demonstrate the development of the axial flow we apply a Helical Wave Decomposition (HWD, Moses 1971; Lesieur 2008) to the vorticity field, which decomposes the field into the sum of a right-handed and a left-handed component: $\boldsymbol{\omega}(\mathbf{x}) = \boldsymbol{\omega}_R(\mathbf{x}) + \boldsymbol{\omega}_L(\mathbf{x})$. The HWD has been applied before in studies of helical turbulence (Q. Chen, S. Chen, and Eyink 2003). Here we follow the approach of previous studies on vortex dynamics using the HWD (Melander and Hussain 1993; Virk, Melander, and Hussain 1994). If the vorticity field is unpolarized, there is no preferred handedness in the vortex lines and the left- and right-handed components of the vorticity field will have equal magnitudes. Following Virk, Melander, and Hussain (1994), a scalar measure of the local handedness of the vorticity field is

$$R_a(\mathbf{x}) = \log_2\left(\frac{|\boldsymbol{\omega}_R|}{|\boldsymbol{\omega}_L|}\right),$$

where $\boldsymbol{\omega}_R$ and $\boldsymbol{\omega}_L$ are the right- and left-handed components of the vorticity field, respectively.

Figure 4.9 visualizes isosurfaces of R_a. In the quarter of the domain where $x > 0$ and $z > 0$, at $t = 4.5$ we observe a structure of negative R_a at the reconnection region, corresponding to the polarization of the vortex lines drawn in figure 4.5. An axial flow away from the bridges confirms our analysis in the previous section. Between $t = 4.5$ and $t = 5.5$ an

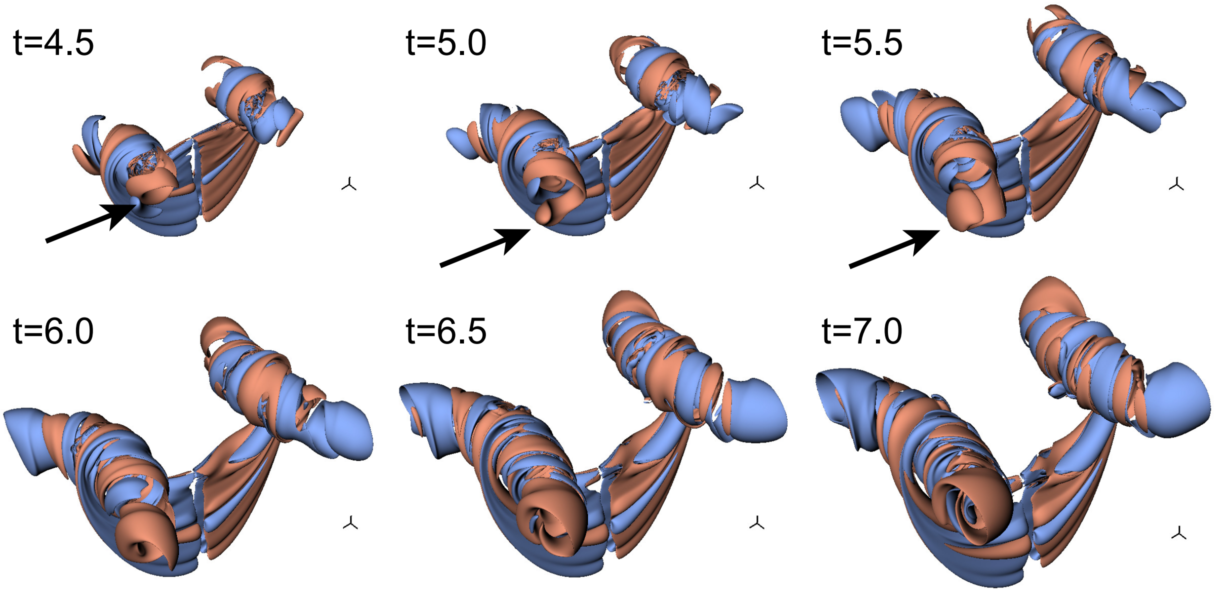


Figure 4.9: Isosurfaces of R_a between $t = 4.5$ and $t = 7.0$. Each isosurface is colored by its sign (positive dark (red online), negative light (blue online)) and their values are $R_a = \pm 0.5$. The arrows at $t = 4.5$–5.5 indicate the right-polarized structure that expands along the tube.

isosurface of positive R_a (indicated by a black arrow at $t = 4.5$–5.5) grows helically from the connection region. This structure corresponds to a dominant right-handed vorticity field which imparts an axial flow towards the bridge region, as explained in the middle panel of figure 4.8. For $t \geq 6.5$ (figure 4.9) a left-handed structure emerges in the outer end of the vortex core at $x > 0$, $z > 0$, with an axial flow away from the reconnection region, as sketched in the right panel of figure 4.8 and visualized by the helicity density in figure 4.10.

During the quiescent phase we observe a secondary-level vortex collision between the threads (see figure 4.11). A third thread-like structure is visible, emanating from the bridges and wrapping around the main threads. The vorticity field reminds one of the connection between two tubes of unequal strength (Marshall, Brancher, and Giovannini 2001; Ortega, Bristol, and Savas 2003). However, because of the low local Reynolds number of the threads ($Re_\Gamma \approx 550$) , the vorticity in the threads diffuses without any significant circulation transfer. One may argue that a higher initial Re_Γ will increase the local thread circulation so that a reconnection between the threads could be enabled. We note however that here the remnant circulation relative to the initial circulation is lower than that observed for vortex tube collision at $Re_\Gamma = 1000$ (Melander and Hussain 1988), so that the thread Re_Γ may increase only slowly with increasing initial Reynolds number

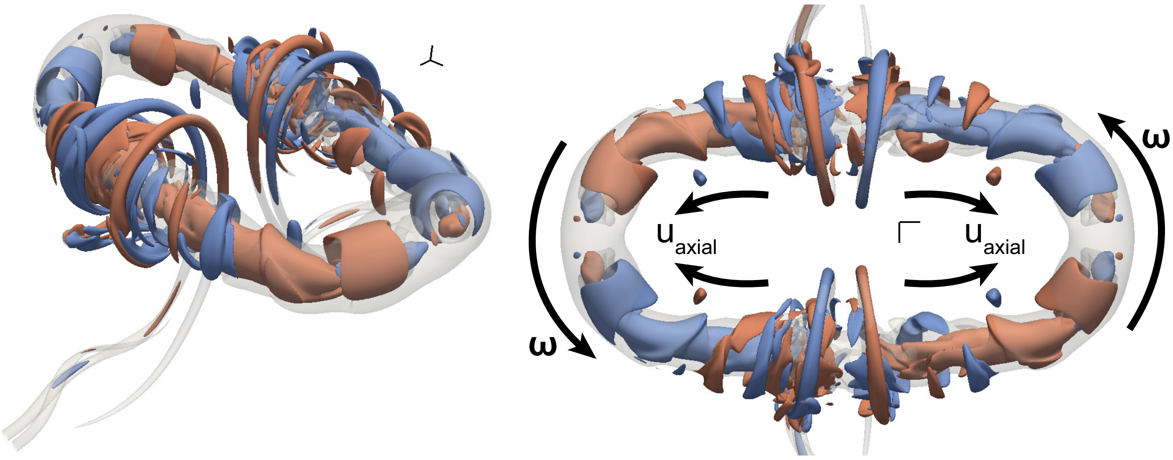


Figure 4.10: Isosurfaces of positive (dark gray, red online) and negative (light gray, blue online) values of $\boldsymbol{\omega}\cdot(\mathbf{u}-\mathbf{U})$ at $t=7.0$, for the same value as figure 4.4, plotted with a periodic image to show one full ring. We show a perspective view (left) and a top view (right), where we annotated the visualization with directions of vorticity and axial flow. A transparent isosurface of $|\boldsymbol{\omega}|=0.5\omega_0$ is shown in gray.

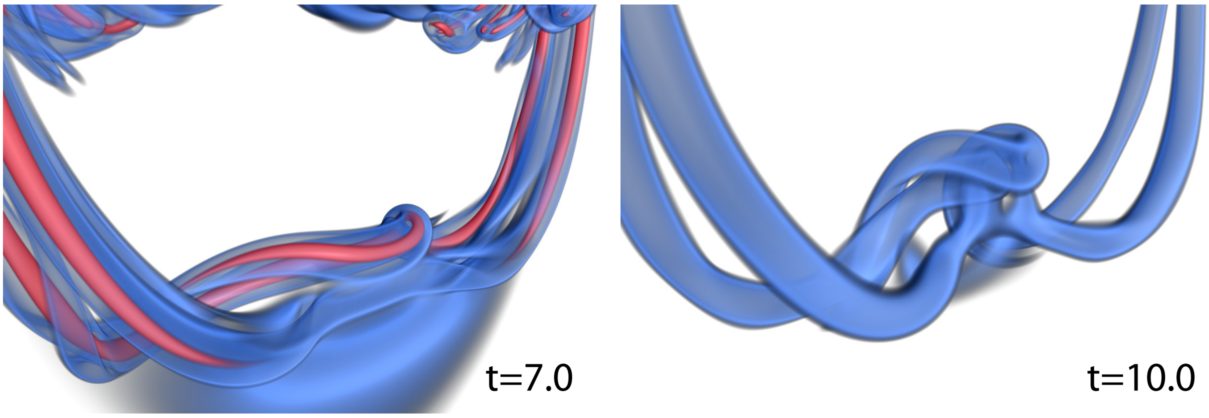


Figure 4.11: Volume renderings of vorticity magnitude showing the remnant threads from the first connection at the specified times. Blue and red are centered around values of $0.14\omega_0$ and $0.8\omega_0$, respectively.

Furthermore, as observed also in Hussain and Duraisamy (2011), the threads are sensitive to a Kelvin-Helmholtz instability due to their stretching by the bridges, so that they are more likely to break-up earlier at higher Reynolds numbers.

4.3.3 Second collision

After their build up and initial oscillatory evolution, the elliptical rings approach each other and collide at $t = 12$. The collision is associated with a burst of small scale vortical structures with large vorticity magnitudes in the region where the two rings meet. During this collision, circulation is transferred by the onset of a reconnection. However, after about 40% of the circulation has been transferred, the process stops at $t = 13.6$. We note that the second collision starts almost simultaneously with the new oscillation cycle of the rings. The ends of the rings moving away from each other due to axis-switching would explain the shorter and smaller circulation transfer.

The energy spectrum during and after this second collision (figure 4.3, right) exhibits a power-law regime, now with a $-7/3$ slope present over more than a decade of wavenumbers from $t = 14.5$ to $t = 22.0$. We report further on the energy spectrum in section 4.4.3, where we relate it to our observations from vortex collision with initial axial flow presented in section 4.4.

For $t > 14$ we observe two large clusters of small-scale structures that move apart, similar to the motion of the bridges in the first collision. Two vortex tubes connecting these clusters are visible, surrounded by small-scale vorticity generated during the last collision. These tubes remind one of the threads containing the remnant circulation after the collision between the initial tubes. However, we note that since the vortex tubes are the topologically new structures, the word ‘threads’ should actually refer to the large-scale structures. These large clusters are expected to undergo further oscillations and further collisions as they evolve, transferring energy to smaller scales with each collision as the flow evolves towards a turbulent state.

4.4 Results with initial axial flow

To study the effect of an initial axial flow in the tubes, we simulated the flow with two different initial swirl numbers: $q = 0.383$ (*case 1*) and $q = 0.958$ (*case 2*). Here q is defined as $q = u_0/u_{\theta,0}$, where u_0 is the prefactor in equation (4.6), corresponding to the peak axial flow strength in the tube, and $u_{\theta,0} = \|\mathbf{u}_{(t=0)} \cdot \mathbf{e}_\theta\|_\infty$ is the peak azimuthal velocity of an unperturbed tube at $t = 0$. We discuss case 1 and case 2 separately, and refer to the simulation without initial axial flow as *case 0*. Flow visualizations of both case 1 and case 2 are presented in figure 4.12.

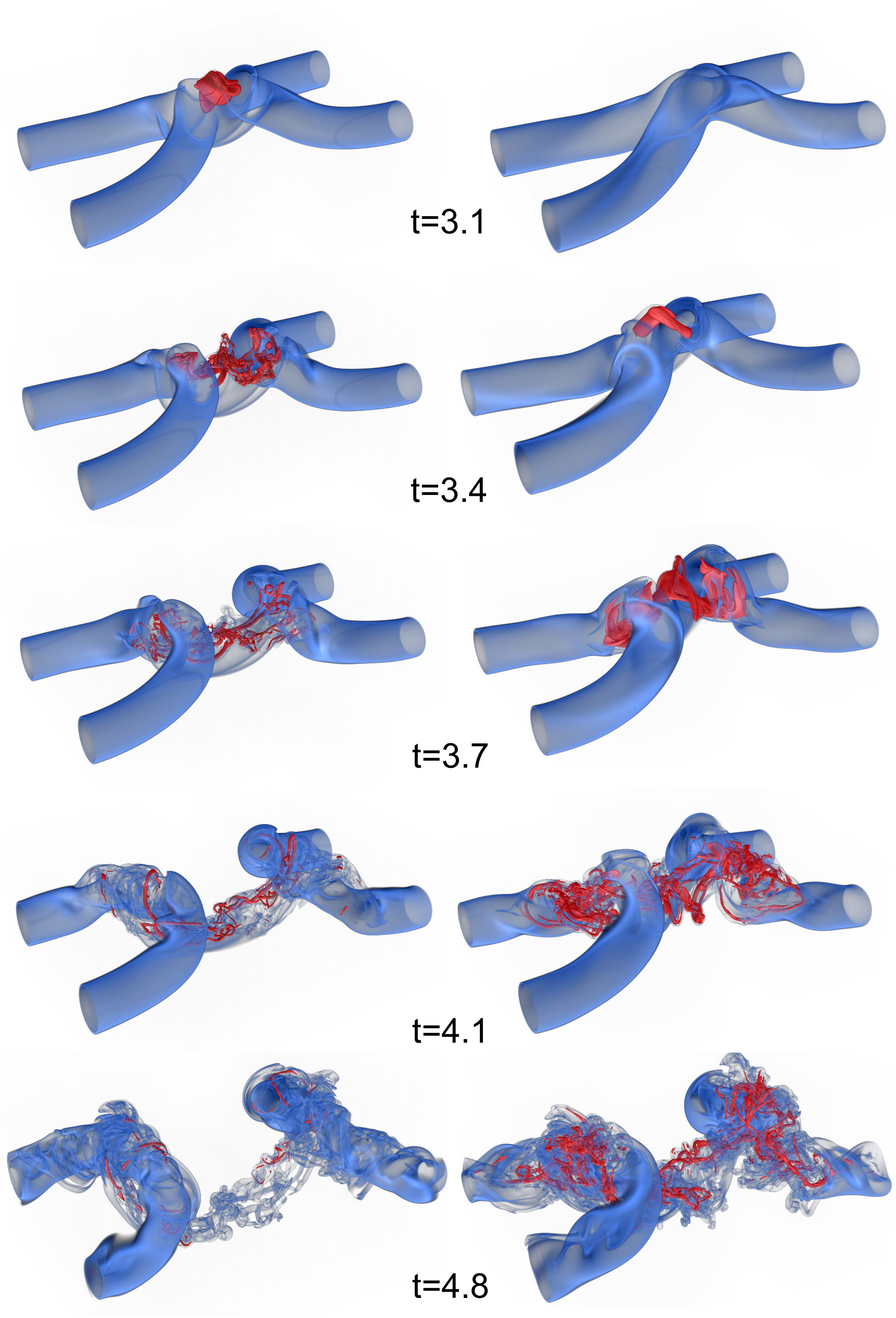
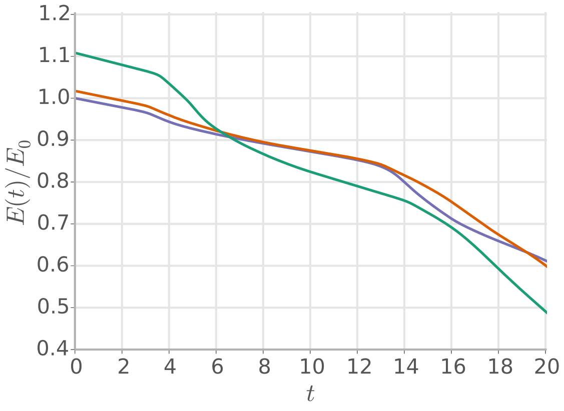


Figure 4.12: Volume rendering of vorticity magnitude for case 1 (left column) and case 2 (right column) at the specified times. The colormap is the same as in figure 4.1.



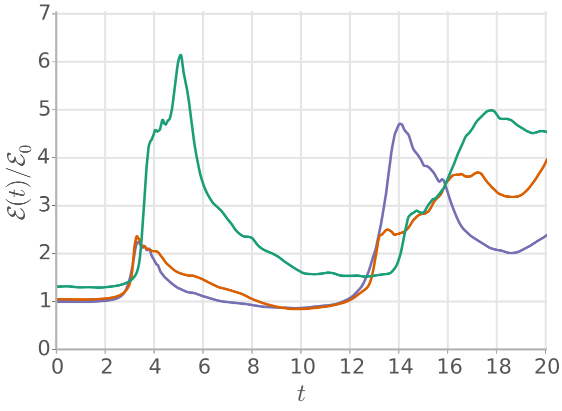
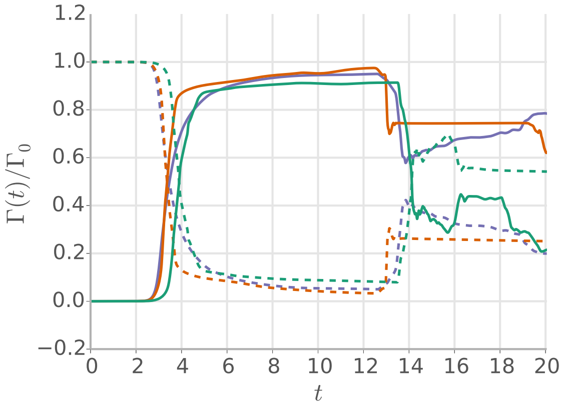


Figure 4.13: Evolution of energy (left) and enstrophy (right) over time, normalized by their initial values for the case without initial axial flow. Case 0 (blue), case 1 (red) and case 2 (green).



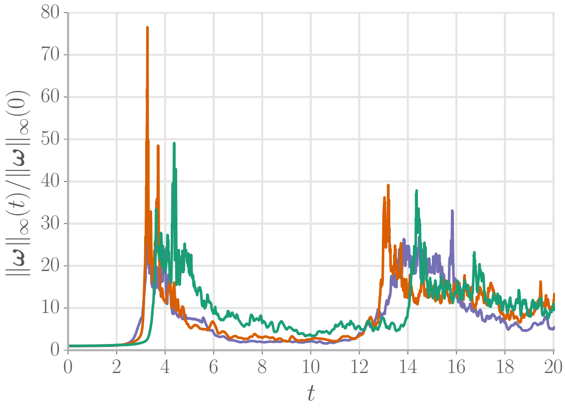


Figure 4.14: Left: Circulation in half of the $x = 0$ (solid lines) and $z = 0$ (dashed lines) planes as a function of time, normalized by the circulation in the $z = 0$ plane at $t = 0$. Right: Evolution of maximum vorticity over time, normalized by the vorticity at $t = 0$. Case 0 (blue), case 1 (red) and case 2 (green).

4.4.1 Case 1: q = 0.383

At the onset of the first collision, the tubes flatten in the mid-plane to form two vortex sheets like a plane jet, with each half having an oblique velocity due to the opposing axial flow. In contrast to case 0 (figure 4.1), the vortex sheet now shears and undergoes a Kelvin-Helmholtz instability, accompanied by a large spike in vorticity magnitude and the rapid transfer of energy to smaller-scale vorticity structures. A helical perturbation travels on one half of each of the newly connected structures, convected by the axial flow that is transferred from the initial anti-parallel tubes to the newly created elliptical rings. The remnant circulation of the collision is contained in several small-scale vortical structures. The value of the peak enstrophy is almost unchanged with respect to case 0 (figure 4.13, right), indicating that the main differences in the vorticity fields between these two cases (i.e. with and without axial flow) are of local nature.

We observe a larger rate and a larger amount of circulation transfer in case 1 (figure 4.14, left), whereas its onset is slightly later than in case 0. These differences are quantified by the maximum circulation transfer rate $\mathrm{d}\Gamma^*/\mathrm{d}t$, where $\Gamma^* = \Gamma/\Gamma_0$ and Γ_0 is the circulation in the symmetry plane at $t = 0$, and $t_{\mathrm{d}\Gamma^*/\mathrm{d}t,\mathrm{max}}$, the time of maximum circulation transfer rate (table 4.1). We also show the reconnection time t_R according to the definition given in Hussain and Duraisamy (2011), where $t_R = t_{0.5} - t_{0.95}$: the time needed to reduce the circulation from 95% of Γ_0 to 50% of Γ_0. The maximum circulation transfer for case 1 is 82% higher, and the reconnection time is 25% smaller than case 0.

The energy spectrum during this collision, as well as during the subsequent collision between the elliptical vortex rings (not shown in figure 4.12) exhibits a persistent power-law regime with $-7/3$ slope (figure 4.15) that is further discussed in section 4.4.3.

We mention that in Naitoh et al. (2002) the role of axial flow in the breakup of vortex structures is studied for an experimental vortex ring at Reynolds number 1600. The breakup in that study differs however from the breakup observed in ours (both cases 1 and 2). The breakup in our results is dominated by a Kelvin-Helmholtz instability originating from anti-parallel axial flows, whereas the breakup of the vortex ring in Naitoh et al. (2002) is dominated by waves amplified by the Widnall instability, together with an induced uniaxial axial flow. That is, our breakup is dominated by shear, whereas theirs by compression of vortex lines.

4.4.2 Case 2: q = 0.958

The shearing of the vortex sheet during the first collision observed in case 1 is even more pronounced in case 2, due to the increased axial flow strength. Again the sheet undergoes rapid instability and the small-scale vortical structures get rapidly advected downstream. This can be seen in figure 4.12 as clusters of small-scale vorticity structures swirling around each of the main tubes. The corresponding smaller localization of strong vorticity is visible in the diminished value of the maximum vorticity of case 2 with respect to case 1 (figure 4.14, right), and the much higher enstrophy values (figure 4.13, right).

Table 4.1 shows that the reconnection time and the maximum circulation transfer rate for case 2 are between those for case 0 and case 1. The onset time for reconnection as well as

Table 4.1: Comparison of several characteristic reconnection quantities with varying axial flow strength.

flow case	$\max \mathrm{d}\Gamma^*/\mathrm{d}t$	$t_{\mathrm{d}\Gamma^*/\mathrm{d}t,\max}$	$t_{0.95}$	$t_{0.5}$	t_R
0	1.042	3.07	2.86	3.46	0.60
1	1.901	3.20	2.91	3.37	0.45
2	1.649	3.63	3.37	3.89	0.52

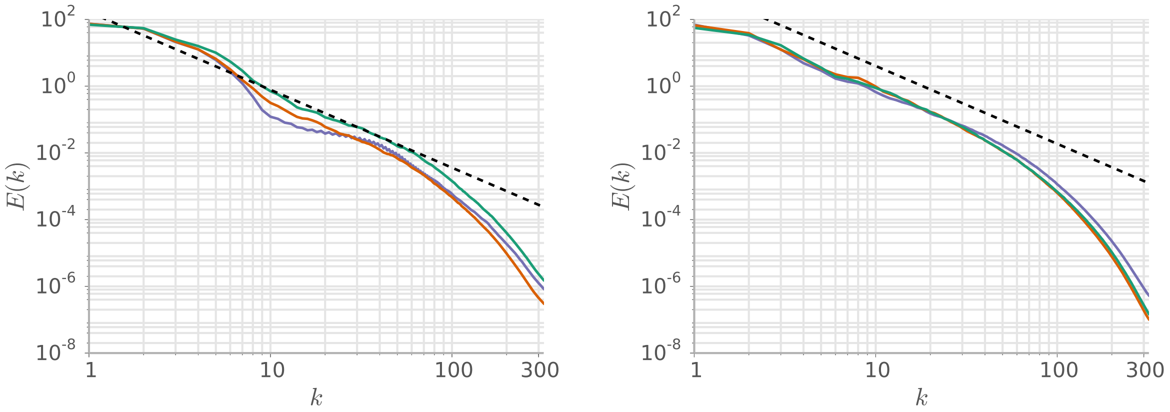


Figure 4.15: Three-dimensional energy spectrum of case 0 (blue), case 1 (red) and case 2 (green) at $t = 3.8$ (left) and $t = 15.5$ (right). The dashed line in both plots corresponds to a $-7/3$ slope.

the time of maximum circulation rate are both delayed with respect to the previous cases. Furthermore, figure 4.14 (left plot) shows that the circulation transfer is smaller than for both case 0 and case 1, as the circulation transfer flattens out at $t \approx 5$.

Again we find a power-law regime in the energy spectrum with $-7/3$ slope (figure 4.15), which will be further discussed in the next section.

4.4.3 Energy spectrum

The $-7/3$ energy spectrum characterizes the second collision of vortex reconnection without any initial axial flow (case 0) (figure 4.3, right), as well as the first and second collisions of vortex reconnection, for both cases 1 and 2, with imposed initial axial flow(figure 4.15). In both cases the common characteristic is the presence of axial flow during the collision of the vortex structures.

We remark that the $-7/3$ energy spectrum has been associated with a cascade of energy dictated solely by the rate of helicity transfer, η. In Brissaud et al. (1973), based on dimensional analysis, it is argued that in a flow with zero energy dissipation rate ε and non-zero helicity dissipation rate η, the spectra of energy and helicity in the inertial range have the form

$$E(k) \sim \eta^{2/3} k^{-7/3}, \tag{4.10}$$

$$H(k) \sim \eta^{2/3} k^{-4/3}. \tag{4.11}$$

This is referred to as the pure helicity cascade or direct helicity cascade, and is contrasted with the case of a joint cascade of energy and helicity, where both η and ε are nonzero, such that

$$E(k) \sim \varepsilon^{2/3} k^{-5/3}, \tag{4.12}$$

$$H(k) \sim \eta \varepsilon^{-1/3} k^{-5/3}. \tag{4.13}$$

Numerical computations have not shown the existence of the pure helicity cascade in isotropic homogeneous turbulence, instead the simultaneous cascade of energy and helicity has consistently been found (Borue and Orszag 1997; Q. Chen, S. Chen, and Eyink 2003; Mininni and Pouquet 2010). In an experimental study on vortex bursts in a laminar flow (Cuypers, Maurel, and Petitjeans 2003), the time-averaged velocity spectrum downstream from the burst region seems to develop a slope close to $-7/3$ (figure 4 in that reference), although this development is not mentioned by the authors.

For all three simulations discussed here, we have plotted the helicity $H(t) = \int_V \mathbf{u} \cdot \boldsymbol{\omega} \, \mathrm{d}V$ in figure 4.16. For vortex reconnection without initial axial flow, the helicity deviates from zero after $t \approx 13$, which coincides with the time when the $-7/3$ slope first appears. In order to examine this correlation we have repeated the case 0 computation in one-fourth of the computational box, using symmetric boundary conditions to ensure zero helicity at all times, and still find the $-7/3$ slope in the energy spectrum during the second collision.

We computed the helicity spectrum for cases 1 and 2 during the first collision. For case 0, the helicity is close to zero; also, the helicity spectrum integrates to a value close to zero. The definition of an inertial range for the helicity spectrum is thus not meaningful in this case. For cases 1 and 2 the helicity is positive and the spectrum at $t = 3.8$ is shown in figure 4.16 (right). Due to oscillations in the spectrum, the slope can not be conclusively identified: for case 1 the slope is around -3 whereas for case 2 the slope can be interpreted both as $-4/3$ or $-5/3$.

We conclude this discussion by noting that in our simulations, the energy dissipation rate is $\varepsilon > 0$ at all times, which should exclude the direct helicity cascade possibility (Brissaud et al. 1973). The results and the above discussion do not conclusively explain the $-7/3$

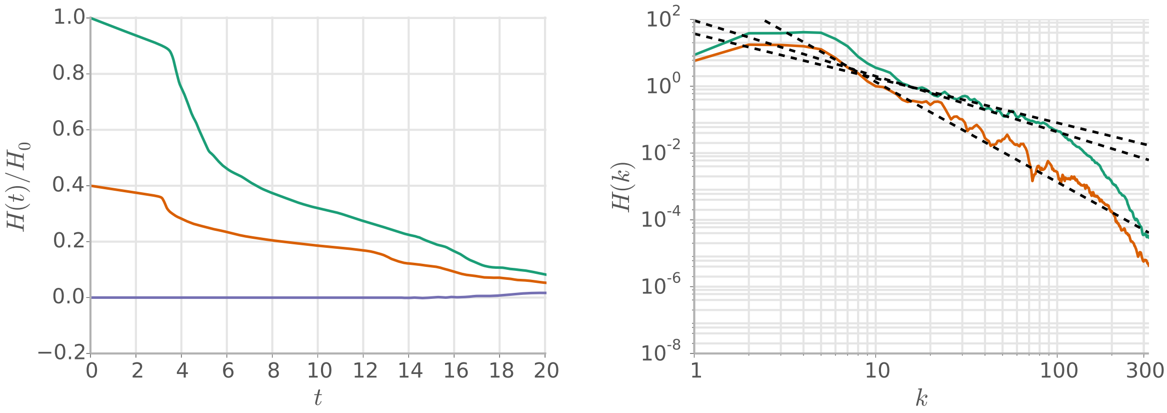


Figure 4.16: Helicity for all three cases, normalized by its value at $t = 0$ for case 2 (left). Helicity spectrum at $t = 3.8$ for cases 1 and 2, black dashed lines are $-4/3$, $-5/3$ and -3 slopes (right). Case 0 (blue), case 1 (red) and case 2 (green).

energy spectrum slope and its relationship to helicity in turbulent flows, which remains subject for further study.

4.5 Conclusions

We have performed long time Direct Numerical Simulations of the reconnection of two anti-parallel vortex tubes at $Re_\Gamma = 10\,000$ with and without initial axial flow. The collision process is described through visualizations of the vortex structures as well as the corresponding energy spectra. The present simulations demonstrate the imparting of an axial flow on the vortex structures formed after the first collision, that largely influences the subsequent collision that is here revealed for the first time.

Vortex reconnection without initial axial flow exhibits a $-5/3$ slope in the energy spectrum during the first collision, as well as a constant-slope energy decay after the collision — both features commonly associated with turbulent flows. The threads of this collision undergo an interaction of their own which is dominated by diffusion due to their low Reynolds number. The elliptical vortex rings resulting from the first collision undergo a full oscillation due to axis-switching before they collide, with a burst of small scale vortical structures. The repeated vortex collisions exhibit features common to turbulent flows, supporting the notion that reconnection events play an important role in the dynamics of turbulent flows. The present simulations do not show however a vortex reconnection cascade, due to the small circulation of the threads and their susceptibility to instabilities. This observation challenges the suggestion that a vortex reconnection cascade could actually be realized, even at initial ReG much larger than 10 000.

The present simulations elucidate also the role of axial flow in vortex collision dynamics. In the case of non-initial axial flow, upon the first collision, elliptical vortex rings are formed and the coiled vortex lines connecting the rings to the threads impart an axial flow inside their cores, directed away from the connection region. During the evolution of the elliptical vortex rings we observe repeated axis-switching accordingly associated with changes in the direction of the axial flow. This axial flow determines the dynamics of the subsequent vortex ring collision that results in a multitude of strong albeit small scale vortical structures. We wish to stress the notion that reconnection between structures without axial flow leads to structures with axial flow as this may be highly relevant to vortex dynamics of turbulent flows.

Adding an initial axial flow to the vortex tubes, with a swirl number of $q = 0.383$, was found to decrease the reconnection time and increase the rate of maximum circulation transfer, while the main flow structures and global diagnostics remain largely unchanged. Further increasing the swirl number to $q = 0.958$ leads to a rapid breakup of the vortex sheet formed at the onset of the collision. Our simulations show the occurrence of a $-7/3$ energy spectrum for a high Reynolds number vortex collision with non-zero axial flow. The possible relationship of this spectrum to helical turbulent flows and in particular the pure helicity cascade remains subject to further research.

Further work involves studies at higher Reynolds numbers, in order to examine the possibility of a reconnection cascade, using wavelet adapted multi-resolution particle methods.

Acknowledgments This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s70. We wish to thank Jean Favre at CSCS (Switzerland) for his assistance with visualization. Figures 4.1, 4.6, 4.11, and 4.12 were rendered using the VolumePerception software (Rossinelli 2011).

Part II

Fluid-structure interaction and optimization

CHAPTER 5

Equations and methods for FSI

In this chapter we present the equations and our computational approach and software for fluid-solid interaction (FSI). Section 5.1 discusses the relevant equations and boundary conditions for both the fluid and the solid. In section 5.2 the no-slip boundary conditions are replaced by a single extended fluid approach, where the body is represented through a volume force. Section 5.3 shows how the body's response to the fluid is implemented. These implementations are combined into our rVM-FSI approach, and we detail the algorithm and software of the resulting solver in section 5.4. Lastly, in section 5.6 we present details on the optimization algorithm and its coupling to the FSI solver.

Throughout this work we only consider cases where the body is neutrally buoyant, so that the body density equals the fluid density and buoyancy effects are ignored. Without losing generality, we set the uniform density to 1 throughout. Furthermore we do not consider the structural mechanics of the body, instead the body is either rigid or deforming according to an imposed deformation velocity.

This chapter is partially based on Gazzola, Chatelain, et al. (2011), where a detailed validation of the method for one- and two-way FSI can be found.

5.1 Equations

Here we consider a computational domain $\Omega = \Omega_f \cup \Omega_s$, where Ω_f is the fluid domain and Ω_s is the body domain. The interface between the fluid and the body is denoted $\partial\Omega_s$.

The incompressible Navier-Stokes equations govern the fluid domain:

$$\nabla \cdot \mathbf{u} = 0 \qquad \mathbf{x} \in \Omega_f, \tag{5.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \qquad \mathbf{x} \in \Omega_f. \tag{5.2}$$

In a viscous fluid flow the no-slip boundary condition holds on $\partial\Omega_s$, enforcing the fluid velocity ($\mathbf{u}$) to be the same as the body velocity ($\mathbf{u}_s$) through $\mathbf{u} = \mathbf{u}_s$. We decompose this body velocity $\mathbf{u}_s$ into its rigid body components and the (imposed) deformation velocity field, so that

$$\begin{aligned} \mathbf{u}_s(\mathbf{x}, t) &= \mathbf{u}_T(t) + \mathbf{u}_R(\mathbf{x}, t) + \mathbf{u}_{\text{def}}(\mathbf{x}, t) \\ &= \mathbf{u}_T(t) + \boldsymbol{\omega}_s(t) \times (\mathbf{x} - \mathbf{x}_{\text{cm}}(t)) + \mathbf{u}_{\text{def}}(\mathbf{x}, t), \end{aligned} \tag{5.3}$$

where $\mathbf{u}_T$ and $\mathbf{u}_R$ are the translational and rotational body velocity, $\boldsymbol{\omega}_s$ is the angular body velocity, $\mathbf{x}_{\text{cm}}$ is the location of the body center of mass and $\mathbf{u}_{\text{def}}$ is the deformation velocity field. The feedback from the fluid on the body is then described by Newton's equation of motion

$$\frac{\mathrm{d}(M_s \mathbf{u}_T)}{\mathrm{d}t} = \mathbf{F}^H, \tag{5.4}$$

$$\frac{\mathrm{d}(\mathbf{J}_s \boldsymbol{\omega}_s)}{\mathrm{d}t} = \boldsymbol{\tau}^H, \tag{5.5}$$

where M_s is the total mass of the body, $\mathbf{J}_s$ its moment of inertia tensor, and $\mathbf{F}^H$ and $\boldsymbol{\tau}^H$ are the hydrodynamic force and torque, respectively, exerted by the fluid on the body.

The imposed body deformation contains no net translational and rotational components, so that

$$\int_{\Omega_b} \mathbf{u}_{\text{def}} \, \mathrm{d}\mathbf{x} = \mathbf{0}, \tag{5.6}$$

$$\int_{\Omega_b} (\mathbf{x} - \mathbf{x}_{\text{cm}}) \times \mathbf{u}_{\text{def}} \, \mathrm{d}\mathbf{x} = \mathbf{0}. \tag{5.7}$$

However, the deformation velocity field can be non-solenoidal, so that the divergence of the velocity field inside the computational domain is split into

$$\nabla \cdot \mathbf{u} = 0 \qquad \mathbf{x} \in \Omega_f, \tag{5.8}$$

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{u}_{\text{def}} \qquad \mathbf{x} \in \Omega_s. \tag{5.9}$$

When reconstructing the velocity field from the vorticity, this non-solenoidal component is taken into account by considering the Helmholtz decomposition of the velocity field

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi} + \nabla \phi, \tag{5.10}$$

where ϕ is a scalar potential and $\boldsymbol{\Psi}$ is the vector streamfunction introduced in section 2.1. By taking, respectively, the curl and the divergence of the above equation we obtain

$$\Delta \boldsymbol{\Psi} = -\boldsymbol{\omega}, \tag{5.11}$$

$$\Delta \phi = \nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{u}_{\text{def}}. \tag{5.12}$$

We can thus account for the non-solenoidal deformation velocity field by solving an additional scalar Poisson equation for ϕ and adding $\nabla\phi$ to the velocity reconstructed from the streamfunction.

The external domain boundary conditions in the following will always be the free-space boundary conditions, so that

$$\mathbf{u} \to \mathbf{U}_\infty \quad \text{as} \quad |\mathbf{x}| \to \infty. \tag{5.13}$$

5.2 Penalization for solid-to-fluid coupling

To enforce the no-slip boundary condition we use the volume penalization method (Angot, Bruneau, and Fabrie 1999; Coquerelle and Cottet 2008; Gazzola, Chatelain, et al. 2011). In this method the body is represented by a characteristic function $\chi(\mathbf{x})$ on a regular Cartesian grid, such that $\chi(\mathbf{x}) = 1$ for $\mathbf{x} \in \Omega_s$ and $\chi(\mathbf{x}) = 0$ for $\mathbf{x} \in \Omega_f$. The Navier-Stokes equations are then extended into the body and supplemented with a penalization force term that forces the fluid velocity in the body to the body velocity $\mathbf{u}_s$. This gives

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \lambda \chi (\mathbf{u}_s - \mathbf{u}) \quad \mathbf{x} \in \Omega, \tag{5.14}$$

where $\lambda \gg 1$ is the penalization parameter. It was proved in Carbou and Fabrie (2003) that the error in this approach is bounded by $||\mathbf{u} - \mathbf{u}_s|| \leq C\lambda^{-1/2}\,||\mathbf{u}||$. We note that λ has units of (time^{-1}), so that a higher value of λ is associated with smaller temporal scales. For large values of λ the penalization term is therefore a stiff term, that needs to discretized appropriately.

In practical applications the characteristic function χ is mollified to increase the stability and accuracy of the numerical scheme. Unless specified otherwise, in this work we use the following mollification

$$\chi(d) = \begin{cases} 0 & d < -\epsilon, \\ \frac{1}{2}\left(1 + \frac{d}{\epsilon} + \frac{1}{\pi}\sin\left(\pi\frac{d}{\epsilon}\right)\right) & |d| \leq \epsilon, \\ 1 & d > \epsilon, \end{cases} \tag{5.15}$$

where d is the signed-distance to the interface, and ϵ a user-specified smoothing parameter that we set to $\epsilon = 2\sqrt{2}\delta x$, with δx the grid spacing. Recently we found improved accuracy of our results when using the FDMH$_1$ Heaviside discretization of Towers (2009), which has the additional benefit that it does not rely on any smoothing parameter. This mollification technique has been used in chapters 11 and 12, as well as in Rossinelli, Hejazialhosseini, van Rees, et al. (2015).

5.3 Projection for fluid-to-solid coupling

Instead of computing the forces on $\partial\Omega_s$ and evolving Newton's equation of motion, we use a projection method (Patankar and Sharma 2005; Coquerelle and Cottet 2008) that computes the body's translational and angular velocities by enforcing conservation of momentum in the system. This approach relies on a so-called weak coupling, where the body and the fluid are updated in an alternating pattern. Specifically, the fluid equations are evolved for one time step both inside and outside of the body, while the body is kept stationary. At the end of the time step we use the extended velocity field inside the body to compute the linear and angular momenta transferred during the time step

$$\mathbf{u}_T = \frac{\rho}{M_s}\int_\Omega \chi\mathbf{u}\,\mathrm{d}\mathbf{x}, \tag{5.16}$$

$$\boldsymbol{\omega}_s = \rho\mathbf{J}_s^{-1}\int_\Omega \chi\left((\mathbf{x}-\mathbf{x}_{\mathrm{cm}})\times\mathbf{u}\right)\,\mathrm{d}\mathbf{x}. \tag{5.17}$$

This approach avoids the need for force and torque integration over the body surface, a process that is often noisy, and instead integrates over the entire body volume through the χ prefactor.

5.4 Remeshed vortex method with penalization and projection

The penalization and projection techniques are coupled with the remeshed Vortex Method that was described in section 2.3. Since the penalization term introduces a stiff term into the Navier-Stokes equations, we apply a Godunov splitting technique and evaluate each of the right-hand side terms sequentially. The individual steps and the order in which we solve them are given in algorithm 2.

Algorithm 2 Algorithm for the rVM-FSI method

$$\chi^n = \chi(\mathbf{x}^n_{cm}, \boldsymbol{\theta}^n, t^n) \tag{5.18}$$

$$\mathbf{u}^n_{\text{def}} = \mathbf{u}_{\text{def}}(\mathbf{x}^n_{cm}, \boldsymbol{\theta}^n, t^n) \tag{5.19}$$

$$\Delta\phi = \chi^n \left(\nabla \cdot \mathbf{u}^n_{\text{def}}\right) \tag{5.20}$$

$$\Delta\boldsymbol{\Psi} = -\boldsymbol{\omega}^n \tag{5.21}$$

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi} + \nabla\phi + \mathbf{U}_\infty \tag{5.22}$$

$$\mathbf{u}_T = \frac{\rho}{M_s} \int_\Omega \chi^n \mathbf{u}\, \mathrm{d}\mathbf{x} \tag{5.23}$$

$$\boldsymbol{\omega}_s = \rho\, (\mathbf{J}_s(\chi^n))^{-1} \int_\Omega \chi^n (\mathbf{x} - \mathbf{x}^n_{\text{cm}}) \times \mathbf{u}\, \mathrm{d}\mathbf{x} \tag{5.24}$$

$$\mathbf{u}_R = \boldsymbol{\omega}_s \times \left(\mathbf{x} - \mathbf{x}^n_{cm}\right) \tag{5.25}$$

$$\frac{\partial \mathbf{u}}{\partial t} = \lambda\chi \left(\mathbf{u}_T + \mathbf{u}_R + \mathbf{u}^n_{\text{def}} - \mathbf{u}\right) \tag{5.26}$$

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} \tag{5.27}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} \tag{5.28}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nu\Delta\boldsymbol{\omega} \tag{5.29}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u}\ \boldsymbol{\omega}) = 0 \tag{5.30}$$

$$\boldsymbol{\omega}^{n+1} = \boldsymbol{\omega} \tag{5.31}$$

$$\mathbf{x}^{n+1}_{cm} = \mathbf{x}^n_{cm} + \mathbf{u}_T \delta t \tag{5.32}$$

$$\boldsymbol{\theta}^{n+1} = \boldsymbol{\theta}^n + \boldsymbol{\omega}_s \delta t \tag{5.33}$$

The splitting allows for an implicit discretization of the penalization term that removes the stability criterion associated with an explicit discretization. Here we apply implicit Euler to the penalization step (5.26) and replace equation (5.27) to obtain

$$\mathbf{u}_\lambda = \frac{\mathbf{u} + \lambda\delta t \chi^n \left(\mathbf{u}_T + \mathbf{u}_R + \mathbf{u}^n_{\text{def}}\right)}{1 + \lambda\delta t \chi^n}, \tag{5.34}$$

$$\boldsymbol{\omega}_\lambda = \boldsymbol{\omega} + \nabla \times (\mathbf{u}_\lambda - \mathbf{u}), \tag{5.35}$$

where $\mathbf{u}_\lambda$ and $\boldsymbol{\omega}_\lambda$ are the velocity and vorticity after penalization.

The integrals in equations (5.23) and (5.24) are approximated using the midpoint rule. The time integration of the vorticity in the stretching (5.28) and diffusion (5.29) terms are performed using a Runge-Kutta 2 time-stepping scheme. The advection step (5.30) is performed using particles as explained in section 2.2, again with a Runge-Kutta 2 time-stepping scheme. The velocity is not evolved during the stretching and advection steps. Each of the derivatives in equations (5.28), (5.29) and (5.35) is computed using fourth-order central finite differences.

For the uniform resolution simulations described in this part, the Poisson equations (5.20) and (5.21), as well as the derivatives in the velocity reconstruction (5.22) are computed in Fourier space. This involves four scalar FFTs to transform the right-hand sides and six scalar inverse FFTs to transform the solutions $\nabla \times \boldsymbol{\Psi}$ and $\nabla \phi$ back to physical space. The right-hand side of equation (5.20) is computed using fourth-order central finite differences prior to its Fourier transform.

The shape reconstruction performed in equations (5.18) and (5.19) are detailed in Appendix B. In this thesis we only apply planar rotations of the body, nevertheless the extension to 3D rigid-body rotations is given in appendix C and implemented in the multiresolution solver used in part III.

5.5 Software

The above algorithm was implemented as a separate client of the Parallel Particle-Mesh library (PPM, Sbalzarini et al. 2006), based on the code described in section 2.3. The software operates on distributed memory, and so the construction of the swimmer is implemented using a separate Cartesian processor topology. The simulations in this part make use of the dynamic domain-size adaptation described in section 2.3. This is especially relevant in FSI applications, since typically the support of the vorticity field drastically changes throughout the simulation.

5.6 Optimization

In the rVM-FSI algorithm a solid obstacle in the flow is defined through the user-specified functions $\chi(\mathbf{x}, t)$ and $\mathbf{u}_{\mathrm{def}}(\mathbf{x}, t)$. By using suitable parametrization of these functions one can easily generate a variety of arbitrarily shaped and deforming obstacles. An optimization algorithm can then identify a set of parameters which uniquely define an obstacle and maximize a specified cost function evaluated during the simulation.

Since individual cost function values are obtained by performing a numerical simulation, the gradient of the cost function with respect to the parameters is not readily available. For such optimizations that rely only on cost function values, a state of the art algorithm is the Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES) (Hansen and Ostermeier 2001; Hansen, S. D. Müller, and Koumoutsakos 2003). This algorithm is part of the Evolutionary Strategies (ES) family of iterative stochastic optimization algorithms (Schwefel 1975; Rechenberg 1994) that are inspired by the Darwinian concept of natural selection.

In this thesis we use CMA-ES in its multi-host, rank-μ and weighted recombination form (Hansen, S. D. Müller, and Koumoutsakos 2003). At each generation, CMA-ES samples p candidate parameter vectors from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$. The distribution's mean $\mathbf{m}$, overall standard deviation σ and covariance matrix $\mathbf{C}$ are adapted based on past successful parameter vectors, ranked according to their cost function values. We note that de-randomized evolution strategies do not ensure that the global optimum is found. The robustness of CMA-ES is mainly controlled by the population size p (Hansen, S. D. Müller, and Koumoutsakos 2003). Therefore a successful optimization hinges on the balance between a large enough p to find a robust solution, while keeping the computational cost, which scales with the number of cost function evaluations, acceptable.

Some solutions proposed by CMA-ES can be inadmissible from a user point-of-view. In this work we specify bounds on the range of each of the parameters. Whenever one of the parameters in a sampled parameter vector exceeds these bounds, the parameter vector is resampled. If the parameters are within the bounds but the corresponding flow simulation is found to violate some external constraint, the data point is rejected by assigning a high default value to its associated cost function. Since CMA-ES aims to minimize the cost function, such data points are discarded during the subsequent ranking and selection process.

CHAPTER 6

C-start: optimal start of larval fish

In the previous chapter we presented the details of the rVM combined with two-way fluid-solid interaction (FSI). We also detailed CMA-ES, an evolutionary optimization algorithm that has successfully been used in combination with fluid dynamics simulations.

In this and the following two chapters we apply the rVM and CMA-ES reverse-engineering framework to self-propelled swimmers. In this chapter we consider the C-start, a natural escape mechanism observed among many different fishes. We apply our optimization framework to investigate what midline kinematics maximize the escape distance of a natural fish. Results are presented for both the 2D and 3D case.

This chapter is based on Gazzola, van Rees, and Koumoutsakos (2012).

6.1 Introduction

Fish burst accelerations from rest are often observed in predator-prey encounters. These motions may be distinguished into C- and S-starts, according to the bending shape of the swimmer. Predators mainly exhibit S-starts during an attack, while C-starts are typical escaping mechanisms observed in prey (Domenici and Blake 1997). Since the first detailed report by Weihs (1973b), C-start behavior has attracted much attention with studies considering the resulting hydrodynamics and the kinematics of these motions (Domenici and Blake 1997; Budick and O'Malley 2000; U. K. Müller 2004; U. K. Müller, van den Boogaart, and van Leeuwen 2008) in experiments (Tytell and Lauder 2002; Epps and Techet 2007; U. K. Müller, van den Boogaart, and van Leeuwen 2008; Conte et al. 2010) and simulations (Hu et al. 2004; Katumata, U. K. Müller, and Liu 2009).

Several works have suggested that C-start is an optimal escape pattern (Howland 1974;

Weihs and Webb 1984; Walker et al. 2005), but to the best of our knowledge, this has been an observation and not the result of an optimization study. The goal of the present work is to perform an optimization of the fish motion, for a specified zebrafish-like geometry, maximizing the distance travelled over few tail beats by a swimmer starting from rest. The optimization couples the covariance matrix adaptation evolution strategy (Hansen, S. D. Müller, and Koumoutsakos 2003) with two-dimensional and three-dimensional flow simulations using a remeshed vortex method with penalization (Coquerelle and Cottet 2008; Gazzola, Chatelain, et al. 2011).

This work is organized as follows: the geometrical model and motion parameterization of the swimmer is presented in section 6.2 and the optimization procedure and the numerical method in section 6.3. We discuss the flow characteristics for the best parameter sets in two and three dimensions in section 6.4, and summarize our findings in section 6.5.

6.2 Geometrical model and motion parameterization

Inspired by the experimental study of zebrafish larva motion (U. K. Müller, van den Boogaart, and van Leeuwen 2008), we consider a two-dimensional swimmer with a similar shape described by the half width $w(s)$ of the body along its arc-length s. In the three-dimensional case, the geometry is described in terms of elliptical cross-sections of height $h(s)$ and width $w(s)$. The piecewise continuous functions characterizing the half width $w(s)$ and height $h(s)$ used here are

$$w(s)=\begin{cases} w_h\sqrt{1-\left(\dfrac{s_b-s}{s_b}\right)^2} & 0\leq s<s_b,\\ (-2(w_t-w_h)-w_t(s_t-s_b))\left(\dfrac{s-s_b}{s_t-s_b}\right)^3 & \\ +(3(w_t-w_h)+w_t(s_t-s_b))\left(\dfrac{s-s_b}{s_t-s_b}\right)^2+w_h & s_b\leq s<s_t,\\ w_t-w_t\left(\dfrac{s-s_t}{L-s_t}\right)^2 & s_t\leq s\leq L,\end{cases} \tag{6.1}$$

where L is the body length, $s_b=0.0862L$, $s_t=0.3448L$, $w_h=0.0635L$ and $w_t=0.0254L$. In the three-dimensional case, the geometry is described in terms of elliptical cross sections

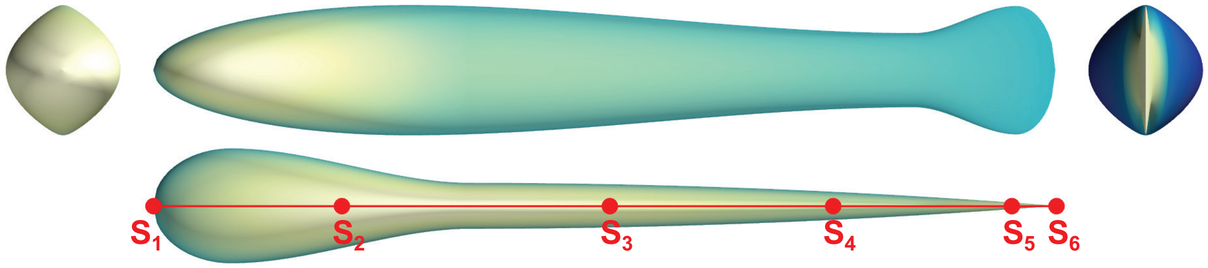


Figure 6.1: Side, top, front and rear view of the implemented geometrical model of the zebrafish larva. On the top view, from left to right, the six curvature control points $S_1 = 0$, $S_2 = 0.2L$, $S_3 = 0.5L$, $S_4 = 0.75L$, $S_5 = 0.95L$, $S_6 = L$ are indicated.

with width $w(s)$ and height $h(s)$, where $h(s)$ is given by

$$h(s) = \begin{cases} h_1\sqrt{1-\dfrac{(s-s_1)^2}{s_1^2}} & 0 \le s \le s_1, \\ -2(h_2-h_1)\left(\dfrac{s-s_1}{s_2-s_1}\right)^3 + 3(h_2-h_1)\left(\dfrac{s-s_1}{s_2-s_1}\right)^2 + h_1 & s_1 < s \le s_2, \\ -2(h_3-h_2)\left(\dfrac{s-s_2}{s_3-s_2}\right)^3 + 3(h_3-h_2)\left(\dfrac{s-s_2}{s_3-s_2}\right)^2 + h_2 & s_2 < s \le s_3, \\ h_3\sqrt{1-\left(\dfrac{s-s_3}{L-s_3}\right)^2} & s_3 < s \le L. \end{cases} \tag{6.2}$$

Here we use the following parameter pairs:

$$(s_1, h_1) = (0.284L, 0.072L), \tag{6.3}$$
$$(s_2, h_2) = (0.844L, 0.041L), \tag{6.4}$$
$$(s_3, h_3) = (0.957L, 0.071L). \tag{6.5}$$

These parameters aim to reproduce the geometry of a 4.4 mm long larva zebrafish of age 5 days post-fertilization (figure 6.1, Fontaine et al. 2008; U. K. Müller, van den Boogaart, and van Leeuwen 2008; Parichy et al. 2009). The motion pattern of a fish performing a C-start is characterized by two phases (Domenici and Blake 1997; U. K. Müller, van den Boogaart, and van Leeuwen 2008). The first phase, denoted the preparatory stroke, consists of the formation of the C-shape. When the fish is fully bent, the second phase involves a propulsive stroke and a backflip of the tail to propel the fish forward. The fast start can then be followed by either continuous swimming or coasting (Domenici and Blake 1997; U. K. Müller, van den Boogaart, and van Leeuwen 2008).

In this study, we employ a parameterization that entails these two phases, leading to a continuous swimming pattern, while the motion parameters are determined by the optimizer. The parameterization is based on the instantaneous curvature κ_s along the mid-line s of the swimmer (Kern and Koumoutsakos 2006), adapted to account for the two phases

$$\kappa_s(s,t) = B(s) + K(s) \cdot \sin\left[2\pi(t/T_{\text{prop}} - \tau(s)) + \phi\right]. \tag{6.6}$$

where $B(s)$ and $K(s)$ are natural cubic splines through the curvature values B_i and K_i at the m interpolation points S_i $(i = 1, \ldots, m)$ along the body of the fish, ϕ is the phase, t is the time, and T_{prop} is the swimming or propulsive period. The phase shift $\tau(s)$ is linearly proportional to the arc-length $\tau(s) = (s/L)\tau_{\text{tail}}$ and is responsible for a traveling wave along the body of the fish.

Hence, the fish starting motion is parameterized by two sets of curvatures, hereafter denoted baseline curvature $\mathbf{B} = \{B_1, \ldots, B_m\}$ and undulatory curvature $\mathbf{K} = \{K_1, \ldots, K_m\}$. The fish starts from an undisturbed position and within the preparatory time interval T_{prep}, transitions to the configuration determined by **B** and **K**. After this stage the preparatory stroke is completed and the fish transitions (within T_{prop}) to the propulsive configuration, completing its first swimming motion cycle. By the end of this cycle, at $t = T_{\text{prep}} + T_{\text{prop}}$, the baseline curvature has returned to zero, and another cycle follows which is solely determined by the curvature **K**. Each transition from one configuration to another is carried out through a cubic interpolation between start and end curvature (setting the first derivative at the extrema to zero to ensure smoothness) within the prescribed time interval. The parameter τ_{tail} is also ramped up via cubic interpolation from zero to its designated value at T_{prep}. We note that the use of a baseline curvature **B** arises from the observation that the C-start motion is not periodic, as it would be if only undulatory curvature **K** were to be considered. Therefore **B** enables the model to capture a broader range of deformations, thus enlarging the reproducible set of possible motions.

We use six control points located at $S_1 = 0$, $S_2 = 0.2L$, $S_3 = 0.5L$, $S_4 = 0.75L$, $S_5 = 0.95L$, $S_6 = L$ (figure 6.1) and the curvature at S_1,S_2 and S_6 is set to zero. These locations and curvature constraints are chosen to model the stiff head, the increasing flexibility towards the posterior of the body and the stiff last fraction of the tail (U. K. Müller 2004; Fontaine et al. 2008; U. K. Müller, van den Boogaart, and van Leeuwen 2008). Furthermore, we fix the ratio between the preparatory and propulsive time interval to $T_{\text{prep}}/T_{\text{prop}} = 0.7$, based on the experimental observations of U. K. Müller, van den Boogaart, and van Leeuwen (2008) and Budick and O'Malley (2000), and set $T_{\text{prep}}+T_{\text{prop}} = 1$ physical time unit. During the optimization we reject cases where $|\kappa_s(s,t)| > 2\pi/L$. These curvature constraints conform with the range of curvature values experimentally observed in fast starts of larval zebrafish (U. K. Müller, van den Boogaart, and van Leeuwen 2008), even though they may not be directly attributed to larval biomechanical properties.

In summary, the fish starting motion is characterized by eight free parameters, namely B_3, B_4, B_5, K_3, K_4, K_5, τ_{tail}, and ϕ, which will be varied during the optimization. We emphasize that biomechanics is not considered in our model and flow-induced body deformations are not accounted for in this study.

6.3 Optimization of starting motion

Based on extensive experimental observations, Domenici and Blake (1997) emphasized that the quantities relevant to evaluating fast starts are distance or speed attained and the time interval of such responses. Therefore we chose to identify the starting motion pattern (characterized by $\mathbf{B}$, $\mathbf{K}$, τ_{tail}, and ϕ) via the optimization of the maximum distance travelled by the center of mass of the swimmer (d^T_{max}) in the time interval $[0, T_{\text{prep}} + 2T_{\text{prop}}]$. Within this interval, the fish performs the preparatory stroke, the propulsive stroke and one additional swimming cycle. This time interval is chosen so as to capture the full start while also allowing examination of the final swimming pattern. The identification of the optimal parameter set is cast into a minimization problem where the cost function is defined as $f = -|d^T_{\text{max}}|$.

The optimization is performed using the covariance matrix adaptation evolutionary strategy (CMA-ES) in its multi-host, rank-μ and weighted recombination form (Hansen, S. D. Müller, and Koumoutsakos 2003; Gazzola, Vasilyev, and Koumoutsakos 2011). Evolution strategies have been shown to be effective in dealing with computational and experimental, single and multi-objective, flow optimization problems (Buche et al. 2002; Kern and Koumoutsakos 2006; Gazzola, Vasilyev, and Koumoutsakos 2011). The CMA-ES is an iterative algorithm that operates by sampling, at each generation, p candidate parameter vectors from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$. Its mean $\mathbf{m}$, overall standard deviation σ and covariance matrix $\mathbf{C}$ are adapted based on past successful parameter vectors, ranked according to their cost function values. We note that de-randomized evolution strategies do not ensure that the global optimum is found. In turn the robustness of CMA-ES is mainly controlled by the population size p (Hansen, S. D. Müller, and Koumoutsakos 2003). Here, as a tradeoff between robustness and fast convergence, we set $p = 100$ for the two-dimensional case, while for the three-dimensional case, given the higher computational costs, we reduced it to $p = 40$.

The search space is bounded for all curvature parameters by $[-2\pi/L, 2\pi/L]$, and for τ_{tail} and ϕ by $[0, 2\pi]$ (furthermore $|\kappa_s(s,t)| \leq 2\pi/L$ at all times and invalid configurations are discarded). Bounds are enforced during the sampling through a rejection algorithm. The initial parameter vector was set to $\mathbf{0}$ for the two-dimensional case, while for the three-dimensional case, we started from the best parameter set found during the two-dimensional optimization. Initial standard deviations were set to 1/15 of the corresponding search space interval.

6.3.1 Equations and numerical method

We consider a deforming and self-propelling body in a viscous incompressible flow determined by solving the incompressible Navier-Stokes equations

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 & \mathbf{x} &\in \Omega_f, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} & \mathbf{x} &\in \Omega_f. \end{aligned} \tag{6.7}$$

where ρ is the fluid density (set equal to body density), ν the kinematic viscosity, and Ω_f the support of the fluid domain. The support of the body domain is denoted Ω_s. The action of the body on the fluid is realized through the no-slip boundary condition at the interface $\partial\Omega_s$, enforcing the body velocity ($\mathbf{u}_s$) to be the same as the fluid velocity ($\mathbf{u}$). The feedback from the fluid to the body is in turn described by Newton's equation of motion $M_s\ddot{\mathbf{x}}_s = \mathbf{F}^H$ and $\mathrm{d}(\mathbf{J}_s\boldsymbol{\omega}_s)/\mathrm{d}t = \boldsymbol{\tau}^H$, where $\mathbf{x}_s$, $\boldsymbol{\omega}_s$, M_s, $\mathbf{J}_s$, $\mathbf{F}^H$ and $\boldsymbol{\tau}^H$ are, respectively, the body's center of mass, angular velocity, mass, moment of inertia and hydrodynamic force and torque exerted by the fluid on the body.

The flow is solved via a remeshed vortex method, characterized by a Lagrangian particle advection, followed by particle remeshing and the use of an FFT-based Poisson solver (Koumoutsakos and Leonard 1995; Koumoutsakos 1997). The remeshed vortex method was coupled to Brinkman penalization,a projection approach handles the feedback from the flow to the body while the body surface is tracked implicitly by a level set (Coquerelle and Cottet 2008). The penalization term added to the Navier-Stokes equations approximates the no-slip boundary condition at the body interface, and allows the control on the solution error through the penalization factor λ. This algorithm has been extended to non-divergence-free deformations for single and multiple bodies (Gazzola, Chatelain, et al. 2011).

Equation 6.7 is cast into its velocity($\mathbf{u}$) - vorticity($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) formulation

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u} : \boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu \Delta \boldsymbol{\omega} + \lambda \nabla \times \chi(\mathbf{u}_s - \mathbf{u}) \qquad \mathbf{x} \in \Omega \equiv \Omega_f \cup \Omega_s, \tag{6.8}$$

where $\nabla \cdot (\mathbf{u} : \boldsymbol{\omega})$ is the vector with components $\partial/\partial x_i(u_i\omega_j)$, χ the characteristic function (Gazzola, Chatelain, et al. 2011) describing the body shape and $\lambda \gg 1$ the penalization factor (here $\lambda = 10^4$ and the mollification length of χ is $\epsilon = 2\sqrt{2}h$, h being the grid spacing).

Translational ($\mathbf{u}_T$) and rotational ($\mathbf{u}_R$) components of $\mathbf{u}_s = \mathbf{u}_T + \mathbf{u}_R + \mathbf{u}_{\text{def}}$ are recovered through a projection approach. The deformation velocity $\mathbf{u}_{\text{def}}$ (prescribed *a priori*) is specific to the shape under study. As $\mathbf{u}_{\text{def}}$ can be non-solenoidal, the incompressibility constraint becomes $\nabla \cdot \mathbf{u} = 0$ for $\mathbf{x} \in \Omega_f$ and $\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{u}_{\text{def}}$ for $\mathbf{x} \in \Omega_s$. This translates into recovering $\mathbf{u}$ from the Poisson equation $\Delta \mathbf{u} = -\nabla \times \boldsymbol{\omega} + \nabla(\nabla \cdot \mathbf{u}_{\text{def}})$, with unbounded boundary conditions (Gazzola, Chatelain, et al. 2011).

During the course of the optimization in two dimensions, simulations were carried out in a domain $[0, 4L] \times [0, 4L]$, with constant resolution 1024×1024 and Lagrangian CFL set to 0.1. Quantities of interest were recomputed at higher resolution (2048×2048). In the three-dimensional case, simulations were carried out in a domain of variable size, growing in time to accommodate the wake. The grid spacing was kept constant at $\delta x = L/256$ during the optimization and at $\delta x = L/512$ for the higher-resolution runs that resulted in the reported diagnostics.

6.3.2 Flow conditions

We define the Reynolds number as $Re = (L^2/T_{\text{prop}})/\nu$, where L is the fish length and T_{prop} is the swimming period. Here we consider a body that models a zebrafish larva of length $L = 4.4$ mm, corresponding to a fish of age 5 days post-fertilization (U. K. Müller, van den Boogaart, and van Leeuwen 2008; Parichy et al. 2009). For a typical swimming period of $T_{\text{prop}} \approx 44$ ms (U. K. Müller, van den Boogaart, and van Leeuwen 2008) in water, we obtain a Reynolds number $Re = 550$ and our simulations were performed at this Reynolds number.

6.4 Results

6.4.1 Two-dimensional swimmer

The course of the optimization for the two-dimensional swimmer is shown in figure 6.2(a) and the best parameter set is presented in table 6.1. This parameter set induces the motion sequence illustrated in figures 6.3 and 6.4. The solution found closely reproduces the starting bout observed experimentally in larval zebrafish (U. K. Müller, van den Boogaart, and van Leeuwen 2008). During the preparatory stroke, the fish bends into a C-shape, then straightens to propel itself forward and completes its cycle.

In order to further quantify C-start mechanics and to elucidate its hydrodynamics, we considered, at time $t_A = T_{\text{prep}} + T_{\text{prop}}$ and $t_B = T_{\text{prep}} + 2T_{\text{prop}}$, respectively, the first (A) and

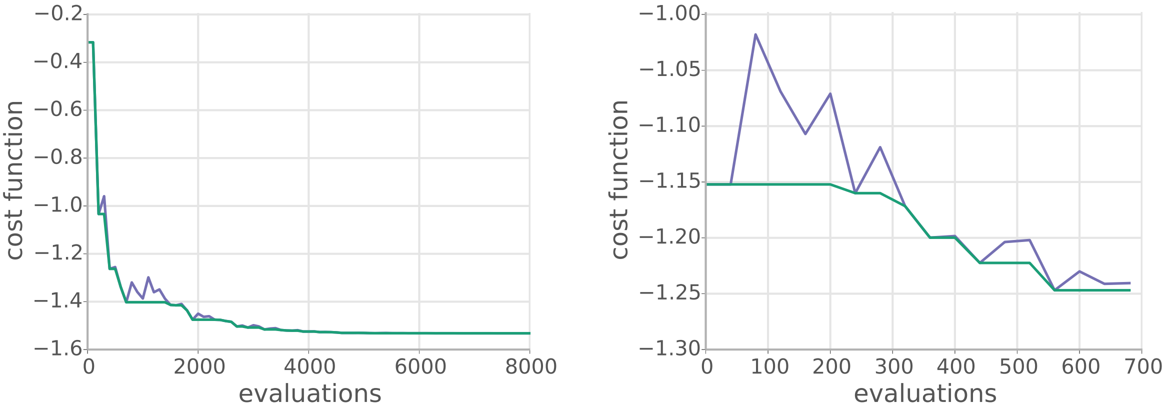


Figure 6.2: Cost function value f (normalized by L) against number of evaluations: two-dimensional (a) and three-dimensional (b) optimizations. Blue and green lines correspond to, respectively, best solution in the current generation and best solution ever. The best two-dimensional solution was used as starting search point in the three-dimensional case.

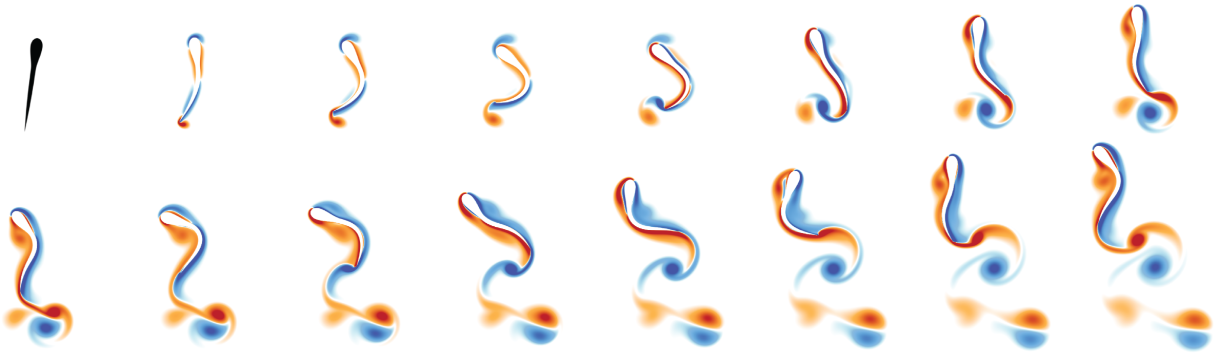


Figure 6.3: Vorticity fields of the two-dimensional best solution (blue negative and red positive vorticity) time sequence ($0 \leq t \leq 2.35 T_{\text{prop}}; \Delta t = 0.156 T_{\text{prop}}$).

second (B) vortex pair generated by nine different motion patterns (figure 6.5). The reference motion pattern is defined by the best parameters as found by the optimization process (case 0). We systematically increased/decreased the curvature $\mathbf{K}^{2D}_{\text{best}}$ and $\mathbf{B}^{2D}_{\text{best}}$ corresponding to the best solutions found (cases −3 to 4 as detailed in figure 6.5) in order to assess the impact of the C-curvature on the flow. We note that cases −3, −2, −1 lie outside the parameter search space of the optimization (since $K_3 > 2\pi/L$). These cases are included in order to explore the effect of curvature values beyond those reported experimentally. Furthermore, in case 5 we considered as reference a “slow start" performed by anguilliform swimming, an archetypal mode of locomotion characterized by the propagation of curvature waves from the anterior to the posterior of the body. We

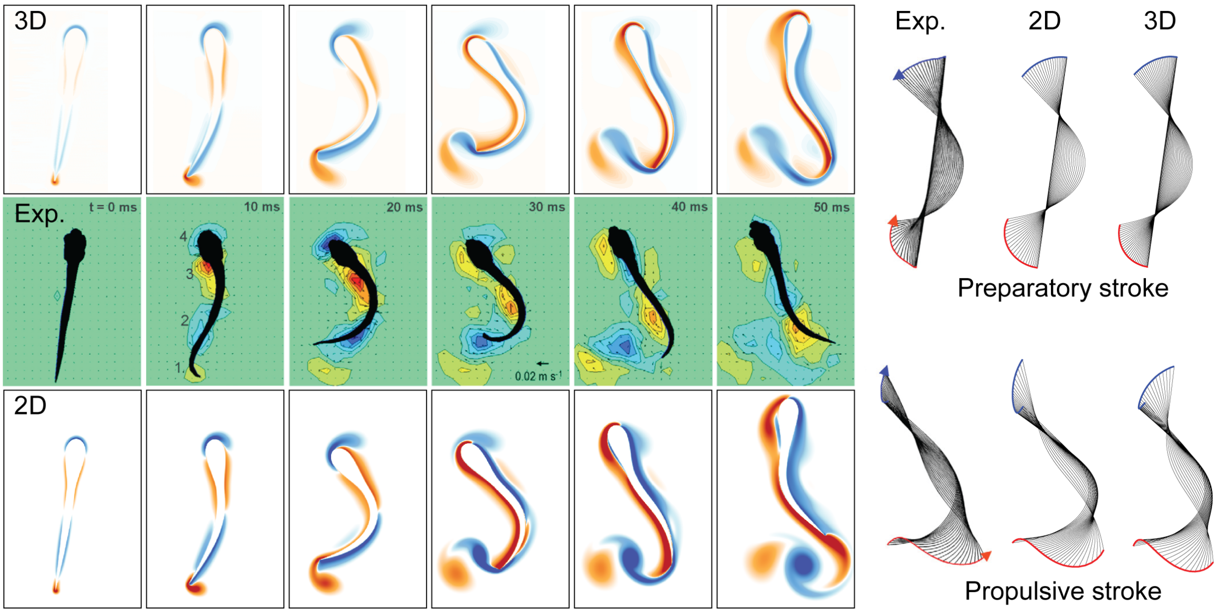


Figure 6.4: Zebrafish larva C-start, experiments and simulations. (left) Vorticity fields (blue negative and red positive vorticity) and (right) swimming kinematics, represented by body mid-lines, corresponding to the best two-dimensional (2D) and three-dimensional (3D) solutions found via the optimization procedure and experimental observations by U. K. Müller, van den Boogaart, and van Leeuwen (2008).

used the kinematic description given in Gazzola, Chatelain, et al. (2011) along with a cubic interpolation to ramp up the motion within T_{prep}. As can be seen in figure 6.5, areas and circulations $|\Gamma|$ of the shed vortex pairs A and B monotonically decrease with reducing curvature magnitude, except for the area of the vortical structure B which shows a maximum for case -1. In case 5 the areas and circulations, of structures A and B, are found to be substantially smaller than in all other cases. Figure 6.5(b) shows that the fish achieves larger speed on increasing the curvature, indicating a correlation between the strength of vortex cores and speed. However, speed does not necessarily translate into a larger distance travelled by the center of mass, due to increased lateral velocities, as depicted in figure 6.5(d) where a maximum is observed for case -1 ($\sim 1\%$ better than case 0). Therefore, the best solution found within the curvature range observed experimentally can be further improved with increased curvature. Nevertheless, within the experimentally observed curvature values we found that C-starts are the best escape mechanisms. The figure also shows the efficiency $\eta = (M_s/2)(\tau^{-1}\int_\tau \mathbf{U}\,\mathrm{d}t)^2/\int_\tau P\,\mathrm{d}t$ (where M_s and $\mathbf{U}$ are the mass and velocity of the swimmer, P is the power imparted to the fluid and $\tau = T_{\text{prep}} + 2T_{\text{prop}}$, see appendix D) of the start procedure. The efficiency peaks between cases 2 and 3, and the anguilliform start outperforms all C-starts. The plot also

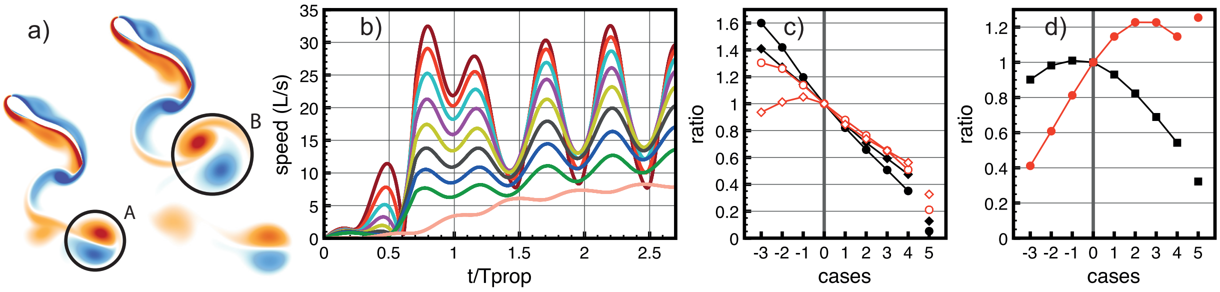


Figure 6.5: Vortex pairs characterization in the two-dimensional setup. Nine motions are considered: the maximum found (case 0), motions with enhanced/reduced curvature with respect to case 0 (130%, 120%, 110%, 90%, 80%, 70%, 60%, respectively, case −3, −2, −1, 1, 2, 3, 4), and (case 5) cruise swimming. a) First (A) and second (B) vortex pairs at time, respectively, $t_A = T_{\text{prep}} + T_{\text{prop}}$ and $t_B = T_{\text{prep}} + 2T_{\text{prop}}$, for case 0. b) Speed of the fish center of mass, from top to bottom, cases −3 to 5, expressed in length/seconds, based on a zebrafish of $L = 4.4$ mm and $T_{\text{prop}} \approx 44$ ms in water. c) Vortex pairs' (A, circle; B, diamond) relative total circulation ($|\Gamma|/|\Gamma_0|$, black) and area ($|A|/|A_0|$, red) with respect to case 0, versus all cases. Vortex cores in A/B, are localized via the criterion $|\boldsymbol{\omega}| > 0.15 \max|\boldsymbol{\omega}_0|$, where $\max|\boldsymbol{\omega}_0|$ is the maximum vorticity of structure A/B in case 0. d) Relative distance travelled (f/f_0, black) and energetic efficiency (η/η_0, red).

shows how C-starts are energetically inefficient, consistent with the observation that burst swimming modes are sustained for short time intervals by fish.

We performed particle tracking for case 0 (figure 6.6), advecting backwards in time passive particles initialized within vortical structure A, to qualitatively illustrate C-start dynamics. Particle locations are superimposed on a forward finite-time Lyapunov exponent (F-FTLE) field (Haller and Yuan 2000) to highlight regions of coherent flow behavior. As can be noticed, particles are accelerated along the body, gaining momentum and circulation before being shed into a vortex pair. The fluid region trapped inside the area bounded by the C-shape of the larva, surrounded by the tracked particles, is accelerated as well and ejected opposite to the swimming direction, without much vorticity being generated. Figure 6.6 pictures the same analysis also for case 3, to illustrate how a smaller curvature corresponds to a smaller fluid region bounded by the C-shape. We conclude that increasing C-curvature results in stronger vortex pairs, which during their formation due to transfer of momentum, contribute to the swimmer's acceleration, albeit at a decreased efficiency.

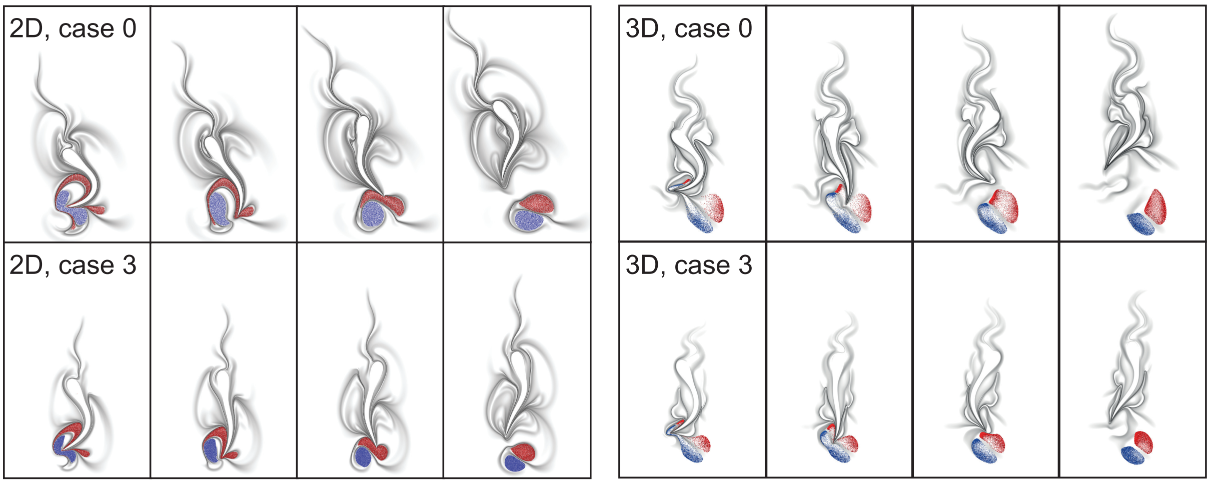


Figure 6.6: Particle tracking and F-FTLE time sequence ($0.5T_{\text{prop}} \leq t \leq 1.25T_{\text{prop}}; \Delta t = 0.25T_{\text{prop}}$). F-FTLE was computed given the integration time $T_{LE} = 1.59T_{\text{prop}}$.

6.4.2 Three-dimensional swimmer

The evolution of the motion parameters for the three-dimensional case is shown in figure 6.2(b), and the best parameter set found is given in table 6.1. The most notable differences between the best parameters in three dimensions with respect to those in two dimensions are the reduced values of B_3 and B_4, indicating a smaller curvature of the posterior half of the fish during the preparatory stroke, and the increased values of K_3, K_4 and K_5, corresponding to larger curvatures during the propulsive stage. Compared to two dimensions, the complete motion in three dimensions differs in the continuous swimming phase, where the large curvature values of K_3–K_5 result in larger deformations of the body (figure 6.7). The forward velocity of the best three-dimensional larval fish is slightly smaller than in two dimensions, and reaches a maximum of 14 L/s during the propulsive stroke and 23 L/s during the ensuing swimming motion. Experimentally reported values for this fish at a similar Reynolds number are 7 L/s for the propulsive stroke and swimming velocities between 14 L/s and 24 L/s (U. K. Müller, van den Boogaart, and van Leeuwen 2008). The simulated fish speeds are higher than those reported experimentally for the propulsive stroke while the speed of swimming motion is in good agreement with the experimental results.

The three-dimensional flow visualizations show that during the preparatory stroke, the larva tail generates a starting vortex ring traveling towards the swimmer's head. The formation of this vortex ring is not accompanied by significant acceleration of the body. During the first backflip of the tail (the propulsive stroke), the fish continuously generates vorticity along the complete profile of the tail. The resulting vortical structure is a vortex

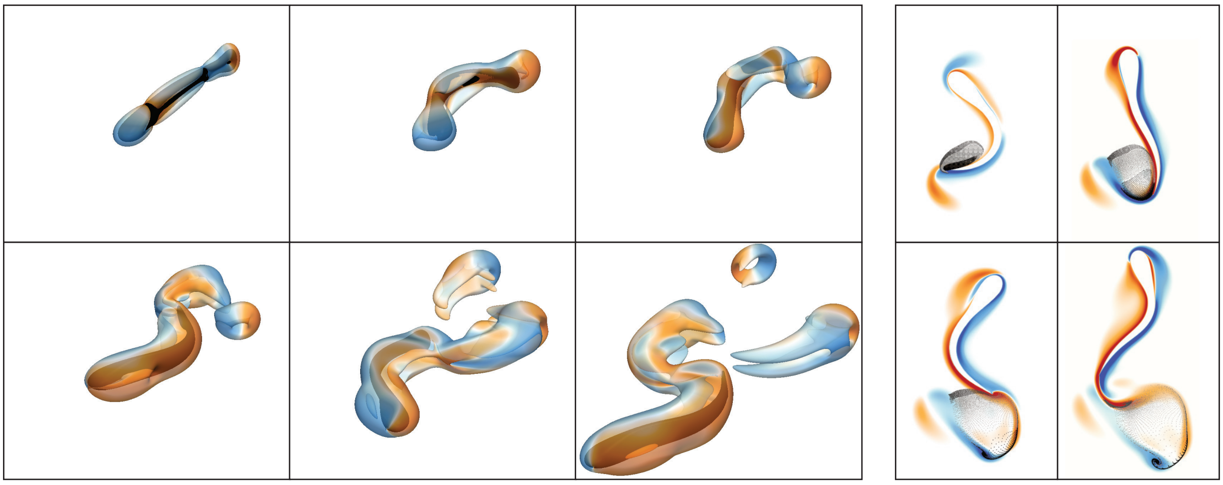


Figure 6.7: Vorticity fields (left) of the three-dimensional best solution found during the optimization (blue negative and red positive z-component of $\boldsymbol{\omega}$) time sequence at times t/T_{prop} 0.13, 0.43, 0.71, 1.04, 1.61 and 2.15 (left to right, top to bottom). Movie available in supplementary material. Evolution of passive tracer particles (right) seeded at $t/T_{\text{prop}} = 0.5$ and overlaid on the ω_z-field. Three sets of particles (dark-medium-light gray-scale) are initialized in different regions of the F-FTLE field. Sequence times as in figure 6.6.

ring elongated along its top and bottom halves (figure 6.7), indicating a stronger vorticity generation at the top and bottom of the tail than at the middle as justified by the tail geometry. The vortical structures generated in subsequent swimming strokes are similar to this elongated vortex ring. The flow structures of the two-dimensional simulations and the mid-plane of the three-dimensional simulations are visually consistent with the experimental results reported in U. K. Müller, van den Boogaart, and van Leeuwen (2008) (figure 6.4). We remark that the elongated vortex rings that can be observed in the experimental results during the propulsive phase are also observed in the vorticity field of the three-dimensional simulations.

We have performed for the three-dimensional simulations a similar analysis as for the two-dimensional case by considering the vorticity field in the horizontal mid-plane, passing through the center of mass of the fish (figure 6.8). We note similar trends for the circulation and area of the vortical structures A and B and fish speed. Also efficiency and distance travelled show a comparable behavior between the two-dimensional and three-dimensional simulations, except for a larger difference in distance travelled between cases −1 and 0 ($\sim$ 4.5%). The particle tracking was performed for the three-dimensional simulation by placing the particles on the first vortex pair in the mid-plane of the swimmer (figure 6.6) and tracing the particles back in time. The particles stay on the mid-plane due to the

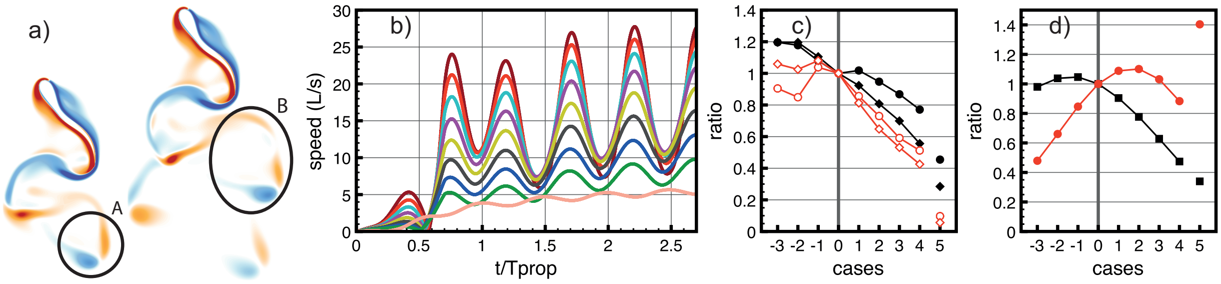


Figure 6.8: As figure 6.5 but for vortex ring characterization in the three-dimensional setup.

	B_3	B_4	B_5	K_3	K_4	K_5	τ_{tail}	ϕ	f
Best 2D	−3.19	−0.74	−0.44	−5.73	−2.73	−1.09	0.74	1.11	−1.53
Best 3D	−1.96	−0.46	−0.56	−6.17	−3.71	−1.09	0.65	0.83	−1.25

Table 6.1: Best motions identified through the optimization (curvatures normalized by L).

vertical symmetry in the flow. However, contrary to the two-dimensional case, only few of the particles trace back to the region bounded by the C-shape of the swimmer. A forward particle tracking was then performed to determine the path of the particles originating in the C-shape region (figure 6.7). The time frames show that the particles spread over a wide region behind the fish, bounded on two sides by a region of vorticity. This indicates that also in the three-dimensional case the fish accelerates a large region of fluid originating within the C-shape of the swimmer but the accelerated fluid spreads out over a much wider region than in the two-dimensional case.

6.5 Conclusions

We have performed a reverse engineering study of larval starting motions to identify the patterns which maximize the escape distance, coupling an evolution strategy with two- and three-dimensional flow simulations. The geometrical model, time scales and parameter search space were dictated by experimental observations of larval fish. The identified best motion kinematics are in good agreement with in *vivo* observations and exhibit C-start patterns, indicating that C-starts do indeed maximize the escape distance. The flow field is characterized by elongated vortex rings stretched by the zebrafish tail. The forward swimming velocities obtained in three-dimensional simulations are in good agreement with those measured in experiments while the vortical structures observed in the mid-plane are consistent with those reported in experiments (U. K. Müller, van den Boogaart, and van Leeuwen 2008). We find that the effectiveness of the C-start escape relies on the ability of pronounced C-bent configurations to trap and accelerate large amounts of fluid. Furthermore C-starts are found to be energetically inefficient. A parametric investigation around our best solutions indicates that increasing curvatures beyond experimentally based bounds can lead to even larger escape distances. Future work will further investigate this latter observation and the multi-objective optimization of motion patterns using multi-resolution flow simulations.

Acknowledgments This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s321.

CHAPTER 7

Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers

In the previous chapter, the optimization framework from chapter 5 was successfully applied to study optimality of the escape start kinematics of larval zebrafish. In the current chapter we study instead optimal morphologies for steady-state swimming with fixed anguilliform midline kinematics. We consider two separate metrics corresponding to fast and efficient swimming. The resulting optimal shapes are analyzed by investigating the sensitivity of our metrics to different morphological features. We provide a detailed study of the midline kinematics to shed light on the general mechanism of undulatory swimming.

This chapter is based on van Rees, Gazzola, and Koumoutsakos (2013).

7.1 Introduction

The striking hydrodynamic performance of natural swimmers (Hertel 1966) has inspired a wide range of engineering applications, from robots to energy harnessing systems (M. S. Triantafyllou and G. S. Triantafyllou 1995; Nachtigall and Bluechel 2000; Bartol et al. 2005). Bioinspired engineering designs often rely on mimicking specific features of natural organisms that are considered to have a functional role (Wilga and Lauder 2004; Nawroth et al. 2012).

In the case of natural swimmers, however, we have limited information to assess which functions are served by morphological features of natural swimmers, and whether such shapes provide an optimal solution. Furthermore, engineering designs may take advantage of available materials and components with flexibilities and restrictions different from those encountered by natural organisms. One may therefore question to what extent engineers should follow a biomimicry approach. In this context the work of Borazjani and Sotiropoulos (2010) demonstrated that manually selected combinations of different shapes and kinematics can swim faster than naturally occurring swimmers. In Gazzola, van Rees, and Koumoutsakos (2012) it was found that C-start escape mechanisms of larval fish can be improved upon by increasing the midline curvature beyond values observed in nature. Fast and efficient swimming shapes and motions for potential flow models have been identified through multi-objective optimization (Tokic and Yue 2012; Eloy 2013).

Here we present shape optimizations with regard to speed and efficiency for self-propelled anguilliform swimmers at moderate Reynolds numbers. We analyze the geometric features that lead to optimality and investigate how these shapes differ from corresponding biological systems.

This work is organized as follows: the shape parameterization and cost functions are presented in section 7.2, followed by a brief review of the employed methodology in section 7.3. The results of the optimization campaigns are analyzed in sections 7.4 and 7.5, and the findings are summarized in section 7.6.

7.2 Physical problem definition

7.2.1 Flow conditions

In this work we study swimmers at a constant Reynolds number based on the fish length L, period of the traveling wave T and dynamic viscosity ν, $Re_{\text{fish}} = (L^2/T)/\nu = 550$ (Gazzola, van Rees, and Koumoutsakos 2012). This flow regime is typical for larval zebrafish 5 days post-fertilization (U. K. Müller, van den Boogaart, and van Leeuwen 2008; Gazzola, van Rees, and Koumoutsakos 2012), which we will use as the biological reference to place into context the optimal solutions.

7.2.2 Midline kinematics

The swimmer's midline kinematics is fixed throughout this work to a motion representative of anguilliform swimming (Carling, Williams, and Bowtell 1998). Anguilliform swimming is an archetypal swimming mode that has been extensively studied, both experimentally (Gray 1933; Tytell 2004b) and computationally (Carling, Williams, and Bowtell 1998; Kern and Koumoutsakos 2006). The larval zebrafish considered here as reference exhibits anguilliform swimming motion (U. K. Müller, van den Boogaart, and van Leeuwen 2008). The swimming motion is based on Carling, Williams, and Bowtell (1998) and Kern and

Koumoutsakos (2006), and is defined by an explicit equation for the lateral displacement of the midline $y_s(s,t)$ in a local frame of reference

$$y_s(s,t) = 0.125L\frac{0.03125 + s/L}{1.03125}\sin\left[2\pi\left(\frac{s}{L} - \frac{t}{T}\right)\right], \tag{7.1}$$

where L is the swimmer's length, s is the arc length of the mid-line of the body ($0 \leq s \leq L$), t is the time, and T the swimming period.

Since our swimmers start from rest, we ramp up their motion by a cubic function during the first cycle, ensuring a smooth transition between the state of rest and the desired motion.

7.2.3 Shape parametrization

The three-dimensional geometry of the swimmer is represented by two orthogonal planar curves for the width and height profile, respectively. Each curve is a bijective, continuous and smooth function of the axial coordinate s. For each $s \in [0, L]$, an elliptical cross-section is created with half-axes based on the width and height curves $w(s)$ and $h(s)$ respectively. The union of cross-sections is the employed doubly-symmetric three-dimensional geometry. We note that, by fixing the length to keep Re_{fish} constant, the volume varies for different shapes.

The height and width profiles are represented by cubic B-splines with N_h and N_w control points (figure 7.1). For both curves the first and last control point are fixed at $(s^w_0, w_0) = (s^h_0, h_0) = (0,0)$ and $(s^w_{N_w-1}, w_{N_w-1}) = (s^h_{N_h-1}, h_{N_h-1}) = (L,0)$, respectively, and serve to maintain C^1 continuity at the end points. The remaining control points are uniformly distributed along the length of the swimmer. Their width and height values, β^w_i and β^h_j with $i \in [1, N_w - 2]$ and $j \in [1, N_h - 2]$, are the parameters to be optimized. We set $N_w = 6$ and $N_h = 8$, for a total of ten parameters $\{\beta^w_i\}_{i=1,\ldots,4}$ and $\{\beta^h_j\}_{j=1,\ldots,6}$ for the optimization.

We remark that in the present simulations the swimming motion is imposed irrespective of the power required to sustain it. Swimmers with large vertical and lateral dimensions will exert correspondingly large forces on the fluid to achieve the prescribed motion. In order to facilitate a comparison with natural swimmers and to generate shapes that are relevant for robotic design, we constrain the search space so that the swimmer fits within a bounding box of size $L \times 0.3L \times 0.3L$. This size of the bounding box is chosen so as to allow a wide range of shapes while remaining relevant to natural swimmers and man-made vehicles.

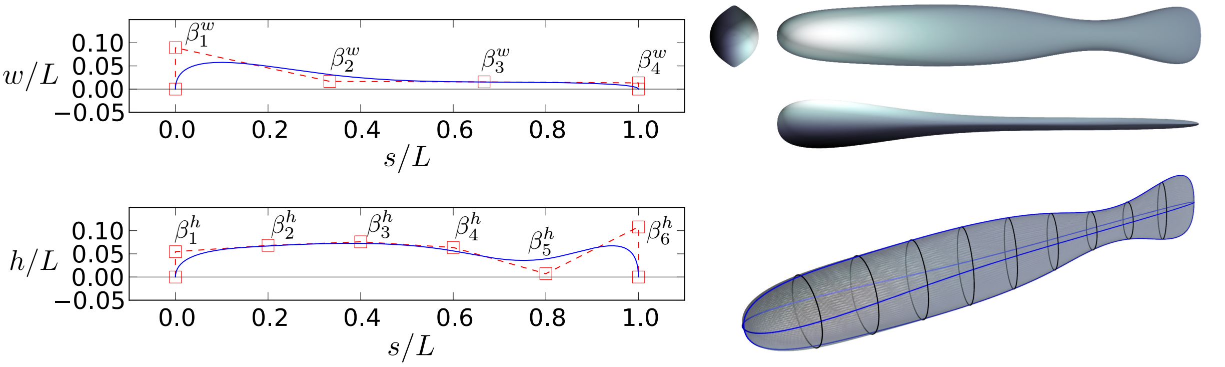


Figure 7.1: Three dimensional shape reconstruction given our parametrization and the vector $\mathbf{m}_0$. The height and width profile (blue curves) are cubic B-splines defined by the locations of the control points β_i^w and β_j^h for each curve (left frame). The 3D shape (right frame) is then generated by constructing the ellipsoidal cross-sections with half-axes defined by the width and height profiles.

7.2.4 Cost function for fast swimming

The fastest shape is found by maximizing the average forward velocity $\overline{U}$ over the sixth swimming cycle, starting from rest. The sixth cycle was chosen as a compromise between reducing computational costs and approximating cruise velocity, based on preliminary studies and the results of Kern and Koumoutsakos (2006). The cost function to be minimized during the optimization is defined as

$$f_{\text{vel}} = -\overline{U} = -\left\| \frac{1}{T} \int_{5T}^{6T} \mathbf{U}(t)\, \mathrm{d}t \right\|_2 , \tag{7.2}$$

where $\mathbf{U}(t)$ is the velocity vector of the swimmer's centre of mass.

7.2.5 Cost function for efficient swimming

The most efficient shape is identified by maximizing, over the sixth cycle, the Froude efficiency, i.e. the ratio between useful energy and the sum of input and useful energy. The input energy is imparted by the swimmer to the fluid through its deformation while the useful energy is the kinetic energy of the swimmer itself. The efficiency cost function is defined as

$$f_{\text{eff}} = -\frac{E_{\text{useful}}}{E_{\text{input}} + E_{\text{useful}}} = -\frac{m\overline{U}^2/2}{\left(\frac{1}{T}\int_{5T}^{6T} P_{\text{input}}(t)\, \mathrm{d}t\right) + m\overline{U}^2/2}, \tag{7.3}$$

where m is the swimmer's mass and P_{input} is the total instantaneous power delivered to the fluid (see appendix D). The swimmer's volume is part of the efficiency definition, that can therefore be understood from an engineering perspective as a *transport* efficiency.

We stress that our objective is to identify optimal swimmer morphologies with respect to relevant engineering metrics. Natural fish have likely evolved under requirements and constraints different from those considered in this study.

7.3 Methodology

7.3.1 Numerical method

We consider a self-propelling body immersed in a three-dimensional viscous fluid governed by the incompressible Navier-Stokes equations

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 && \mathbf{x} \in \Omega_f, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} && \mathbf{x} \in \Omega_f, \end{aligned} \tag{7.4}$$

where Ω_f is the fluid domain, whereas Ω_s denotes the volume occupied by the swimmer. The no-slip boundary condition at the geometry interface $\partial\Omega_s$, matches the fluid velocity $\mathbf{u}$ to the local body velocity $\mathbf{u}_s$. The feedback from the fluid to the body follows Newton's equations of motion

$$\begin{aligned} M_s \ddot{\mathbf{x}}_s &= \mathbf{F}^H, \\ \frac{\mathrm{d}(\mathbf{J}_s \boldsymbol{\omega}_s)}{\mathrm{d}t} &= \boldsymbol{\tau}^H, \end{aligned} \tag{7.5}$$

where $\mathbf{F}^H$ and $\boldsymbol{\tau}^H$ are the hydrodynamic force and torque exerted by the fluid on the body, characterized by centre of mass $\mathbf{x}_s$, angular velocity $\boldsymbol{\omega}_s$, mass M_s and moment of inertia $\mathbf{J}_s$.

The numerical method to discretize and advance equations (7.4) and (7.5) in time consists of a remeshed vortex method, with a penalization technique (Angot, Bruneau, and Fabrie 1999) to account for the no-slip boundary condition and a projection method (Coquerelle and Cottet 2008; Gazzola, Chatelain, et al. 2011) to capture the action from the fluid to the body. The body geometry is represented with a characteristic function χ ($\chi = 1$ inside the body, $\chi = 0$ outside and mollified at the interface) and its motion is defined by the deformation velocity field $\mathbf{u}_{\text{def}}$. Further details, validation and verification of the method can be found in (Gazzola, Chatelain, et al. 2011; Gazzola, van Rees, and Koumoutsakos 2012). In this study, we discretize the domain with a uniform grid spacing of $h = L/256$ during the optimization and $h = L/512$ for the diagnostics reported. The mollification length of χ is set to $\epsilon = 2\sqrt{2}h$, Lagrangian CFL to LCFL $= 0.1$, and penalization factor $\lambda = 10^4$ (Gazzola, Chatelain, et al. 2011).

7.3.2 Optimizer

Optimal shapes (characterized by $\{\beta_i^w\}_{i=1,\ldots,4}$ and $\{\beta_j^h\}_{j=1,\ldots,6}$) are identified by the stochastic optimization algorithm Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) in its multi-host, rank-μ and weighted recombination form (Hansen, S. D. Müller, and Koumoutsakos 2003). CMA-ES has been shown to be robust and efficient in dealing with optimization problems for fluid flows(Kern and Koumoutsakos 2006; Gazzola, van Rees, and Koumoutsakos 2012). Throughout this work we set the population size $p = 40$, which we found to be a good compromise between robustness and computational cost. We initialize our parameters ($\mathbf{m}_0$) to reproduce the geometry of a zebrafish larva at 5 days post fertilization (figure 7.1 and table 7.1).

The search space bounds are enforced through a rejection algorithm. We consider invalid, for numerical reasons, cases in which the vertical coordinate of β_1^w, $\beta_{N_w-2}^w$, β_1^h or $\beta_{N_h-2}^h$ is smaller than the grid spacing used in our simulations. Invalid configurations are rejected by assigning a high default cost function value. Furthermore, we reject cases in which one or both profile curves cross the body midline.

7.4 Results

7.4.1 Optimal shapes

The evolution of the fitness functions, the corresponding optimal shapes and parameters are presented in figure 7.2 and table 7.1. The width profile of the fastest swimmer is reminiscent of a thin airfoil, with a maximum thickness of $0.03L$ and an inflection point in the posterior half. The height profile shows a dorsal and caudal fin, connected by a caudal peduncle. The most efficient swimmer, in contrast, is thicker and blunter than the fastest swimmer with a 4.6 times larger volume and a maximum width of $0.11L$. An inflection point is present in the posterior width. The height profile is characterized by a large lateral area with a smooth transition between the dorsal fin and the caudal fin.

The efficiency, forward and lateral velocities of the fastest, most efficient and zebrafish larva ($\mathbf{m}_0$) shapes are shown in figure 7.3. We note that compared with the shape of the zebrafish larva, the fastest and most efficient shapes achieve a 40% higher velocity and 135% higher efficiency, respectively. These improved performances, however, come at the

	β_1^w	β_2^w	β_3^w	β_4^w	f_{vel}	f_{eff}
$\mathbf{m}_0$	$8.9e^{-2}$	$1.7e^{-2}$	$1.6e^{-2}$	$1.3e^{-2}$	$-3.2e^{-1}$	$-5.7e^{-2}$
fastest	$1.4e^{-2}$	$4.6e^{-2}$	$2.2e^{-3}$	$5.8e^{-3}$	$-5.0e^{-1}$	$-8.5e^{-2}$
most efficient	$7.1e^{-2}$	$1.5e^{-1}$	$1.3e^{-2}$	$7.7e^{-3}$	$-4.3e^{-1}$	$-2.0e^{-1}$

	β_1^h	β_2^h	β_3^h	β_4^h	β_5^h	β_6^h
$\mathbf{m}_0$	$5.5e^{-2}$	$6.8e^{-2}$	$7.6e^{-2}$	$6.4e^{-2}$	$7.2e^{-3}$	$1.1e^{-1}$
fastest	$2.3e^{-1}$	$1.7e^{-1}$	$1.7e^{-1}$	$2.7e^{-1}$	$9.0e^{-2}$	$3.0e^{-1}$
most efficient	$1.6e^{-1}$	$2.6e^{-1}$	$2.8e^{-1}$	$3.0e^{-1}$	$2.3e^{-1}$	$2.1e^{-1}$

Table 7.1: Optimal solutions and start searching point $\mathbf{m}_0$ (approximation of zebrafish larva, Gazzola, van Rees, and Koumoutsakos 2012). The numerical data corresponds to swimmers with $L = 1$, $T = 1$ and $Re_{\text{fish}} = 550$.

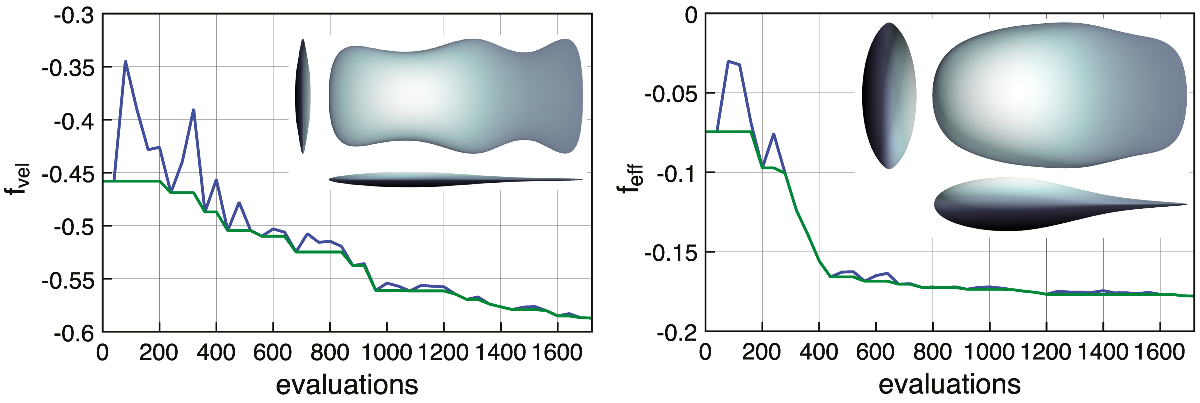


Figure 7.2: Evolution of cost function f_{vel} (left) and f_{eff} (right) against number of cost function evaluations. Blue and green lines correspond to, respectively, best solution in the current generation and best solution ever. The inset of each plot shows the corresponding optimal shape. The numerical data corresponds to swimmers with $L = 1$, $T = 1$ and $Re_{\text{fish}} = 550$, simulated with resolution $h = L/256$ due to computational costs.

expense of a larger input power. In particular, the average power required by the fastest and most efficient solution during the sixth cycle is, respectively, 3.2 and 4.0 times larger than that for the zebrafish larva.

7.4.2 Flow analysis

Both optimal swimmers accelerate during the first half and decelerate during the second half of each stroke. Forward and lateral velocities peak when the tail reaches its maximum

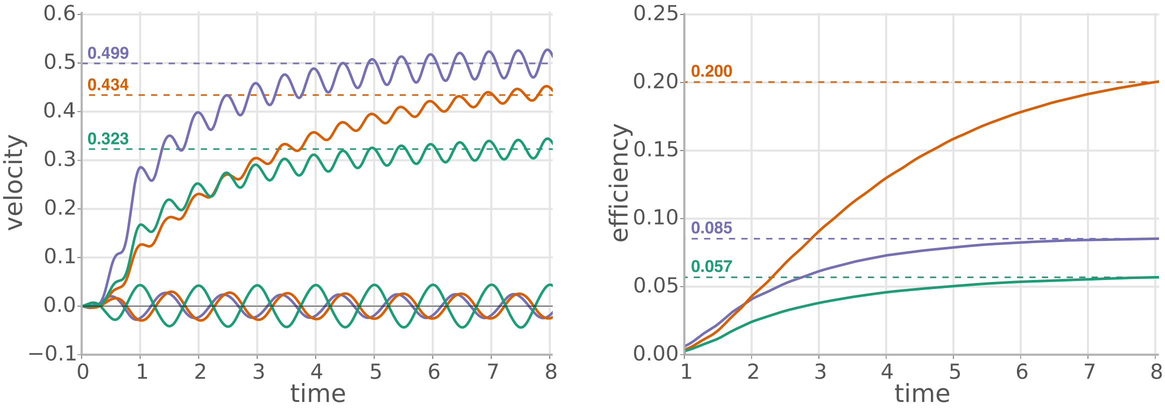


Figure 7.3: Left: forward and lateral velocities, dashed lines denote average velocity in the eighth cycle. Right: efficiency in a running window of length $\Delta T = T = 1$, dashed lines denote efficiency as $t = 8T$. Colours denote the fastest shape (blue), the most efficient shape (red) and the zebrafish larva $\mathbf{m}_0$ (green). The numerical data corresponds to swimmers with $L = 1$, $T = 1$ and $Re_{\text{fish}} = 550$.

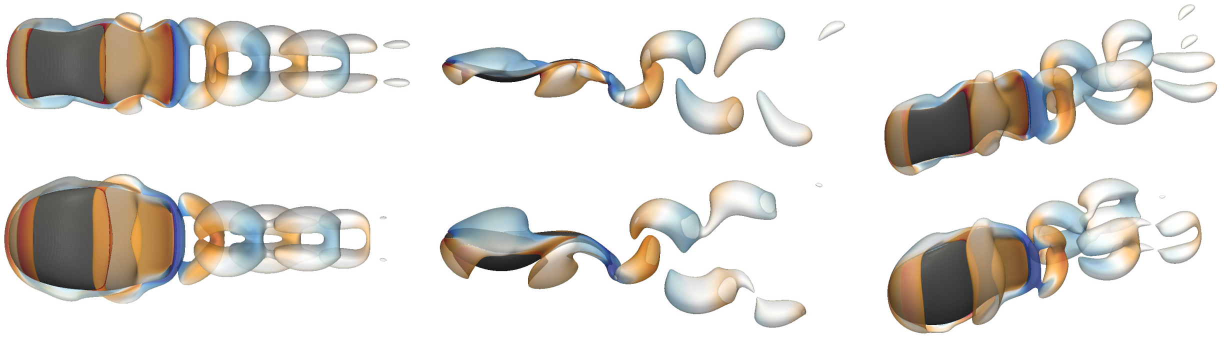


Figure 7.4: The swimmers' (dark gray) wake structure visualized via an isosurface of the Q-criterion, coloured by the vertical component of the vorticity $\boldsymbol{\omega}$. The swimmers are plotted in dark gray. We show a side view (left), top view (middle) and perspective view (right) for the fast (top) and efficient (bottom) swimmer. Animations are presented in the supplementary material.

lateral velocity (i.e. half-way during a stroke, after shedding the upstream leg of a wake vortex ring). The minimum forward velocity is achieved at the beginning of each stroke, when the tail changes direction.

Figure 7.4 shows the structure of the vorticity field for the fastest swimmer visualized by the Q-criterion. In the second half of each tail beat, the downstream half of a lateral elliptic vortex ring is generated, which is completed in the first half of the next tail beat. Each vortex ring has a stronger circulation in the downstream half than in the upstream

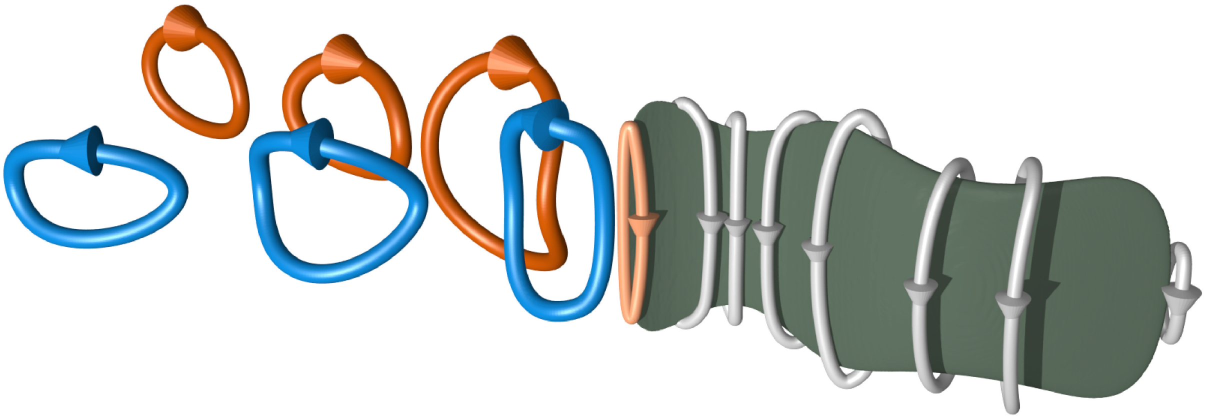


Figure 7.5: Vortex line representation of a selection of vortex rings generated by the fast swimmer. The colors denote their orientation once they have been laterally rotated and shed into the wake.

half. The rings are initially generated with an aspect ratio of approximately 1.9 and then relax to a circular shape. The flow field of the most efficient swimmer (figure 7.4) shows qualitatively the same structures as the fastest swimmer. The lack of a distinct caudal fin results in weaker upstream halves of the vortex rings, so that their asymmetric self-induced velocities cause the rings to rotate outwards around the z-axis. Similar to the fastest swimmer, elliptical vortex rings are generated from the tail with an aspect ratio of approximately 1.7 and relax to a circular shape downstream.

The fluid-solid interaction is driven by the shape-independent midline kinematics. We estimate the vorticity associated with the midline motion as $\partial u_n/\partial\tau$, where u_n is the normal velocity of the midline and τ the tangential coordinate. Visualizing this quantity reveals two midline vortices that move downstream with the traveling wave. During swimming, these midline-induced vortices modulate the boundary-layer vorticity in the midplane (figure 7.6, left). As a result, on either side of the body stronger and weaker vortices alternate and are shed into the wake, forming a 2P vortex street (C. H. K. Williamson and Roshko 1988). The three-dimensional vorticity field reveals that vortex rings are formed in the near flow around the body and travel downstream. These rings are eventually rotated by the body and shed as lateral vortex rings into the wake (figure 7.5).

Steady undulatory swimming is sustained by transferring momentum to the fluid with each stroke of the tail, in order to balance drag-induced decelerations in between strokes. For anguilliform motion, the associated midline vortices modulate the vorticity on either side of the swimmer, creating coherent flow regions (figure 7.6, left). These regions are accommodated within the concave sides of the shape and pushed downstream by the body traveling wave, accelerating the swimmer. We remark that the regions identified

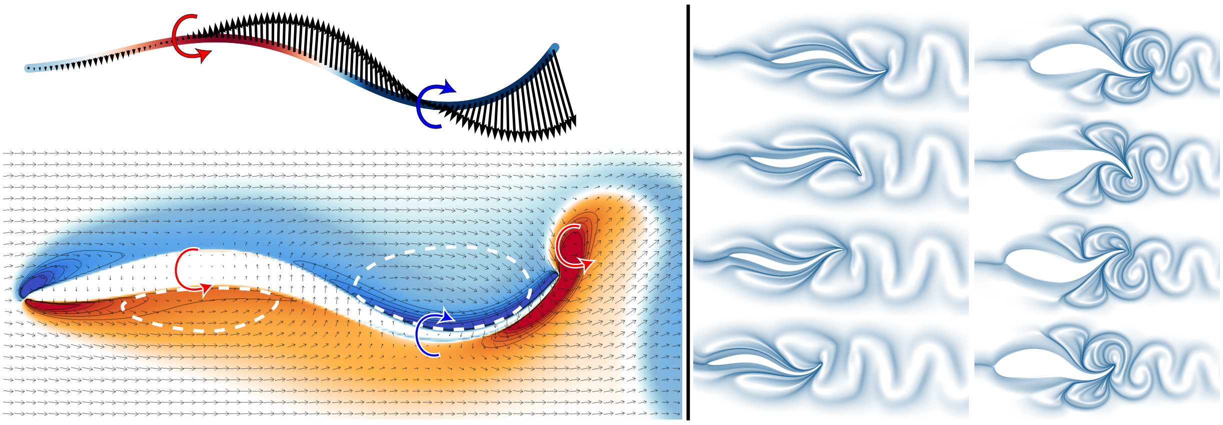


Figure 7.6: Left panel: vortices generated by the midline kinematics (top) modulate the boundary layer vorticity during swimming (bottom, black arrows denote velocity, colors denote vorticity in the midplane, white dashed lines sketch coherent flow regions). Right panel: the F-FTLE field, simulated over $\Delta T_{\mathrm{FTLE}} = 2T$, reveals differences in flow structures for the fast (left) and efficient (right) width profile in a 2D simulation.

through this mechanism (white dashed lines in figure 7.6) correspond to the thrust regions visualized in figure 14 in Reid et al. (2012). The basic mechanism of propulsion through the acceleration of coherent flow regions characterizes also the C-start escape response (Gazzola, van Rees, and Koumoutsakos 2012).

To further elucidate this process, we simulated the optimal width profiles in two dimensions and analyzed the corresponding flow field by means of a forward finite time Lyapunov exponent (F-FTLE), shown in figure 7.6 (right). Two-dimensional simulations, although not able to capture the full physical problem, are instructive in understanding swimming hydrodynamics (Gazzola, van Rees, and Koumoutsakos 2012). For both swimmers, the F-FTLE reveals the above-mentioned regions of coherent flow in an inertial frame of reference, which conform to the body geometry and kinematics and are shed as alternating compact vortices in the wake. Due to the high swimming speed of the fastest swimmer, the flow regions are longitudinally stretched and extend upstream along the entire body. The most efficient but slower swimmer, in contrast, extracts fluid from a less elongated and more circular region along the posterior 25% of the body length. This could be a characteristic of efficient swimming.

7.5 Shape variations

We systematically vary single morphological features of the obtained optimal shapes to study their impact on swimming performance. During this variation we neglect the

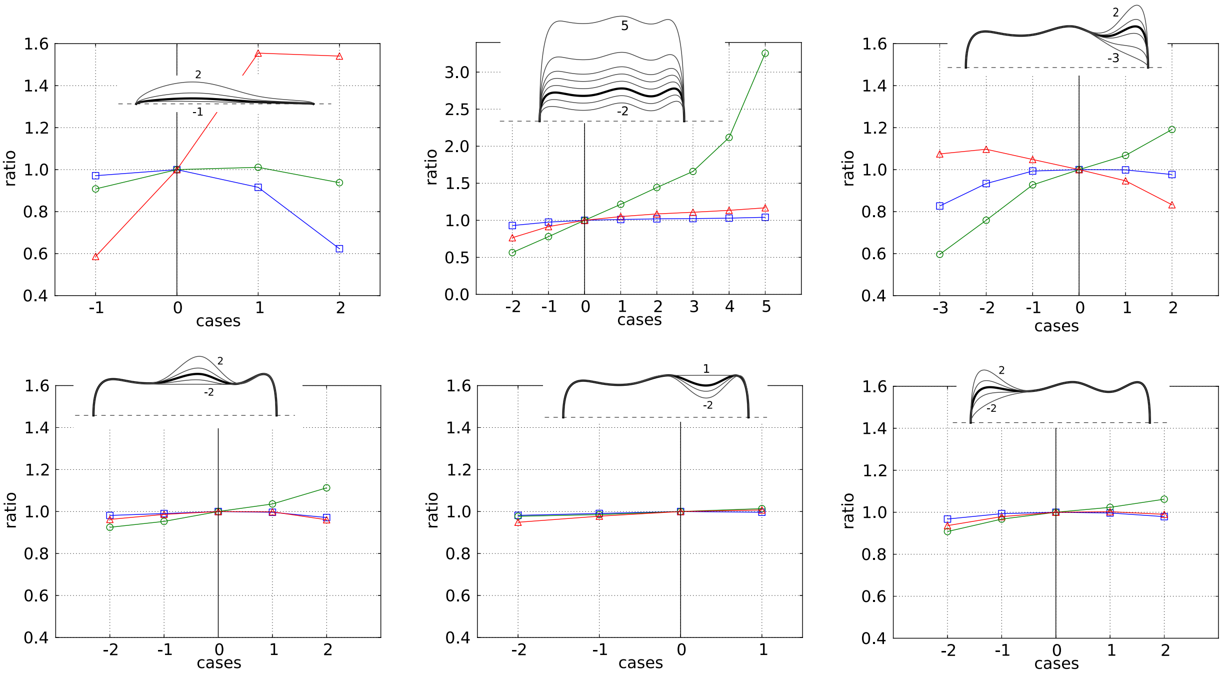


Figure 7.7: Effect of shape variations on the forward velocity (blue squares), efficiency (red triangles) and input energy (green circles) for the fast swimmer.

bounding box imposed on the optimization to detect trends in an unconstrained setting. Besides the forward velocity and efficiency, we show for each variation the power delivered to the fluid integrated over the sixth swimming cycle, as a measure of the associated effort, in Figs. 7.7 and 7.8.

7.5.1 Width

Increasing the width for the fast swimmer shows a 40% drop in forward velocity, and a dramatic improvement (up to 60%) in efficiency. For the efficient swimmer we also find that thinner shapes improve speed and degrade efficiency. Thicker shapes, on the other hand, require a larger input energy than our most efficient swimmer, which degrades their efficiency.

We find that the width profile has the largest influence on the swimmers' performance. A thin shape maximizes the forward velocity by minimizing the pressure drag along the bulk of the body, whereas a thicker width profile is beneficial for efficiency by striking an optimal balance between energy losses in the wake and forward momentum of the swimmer.

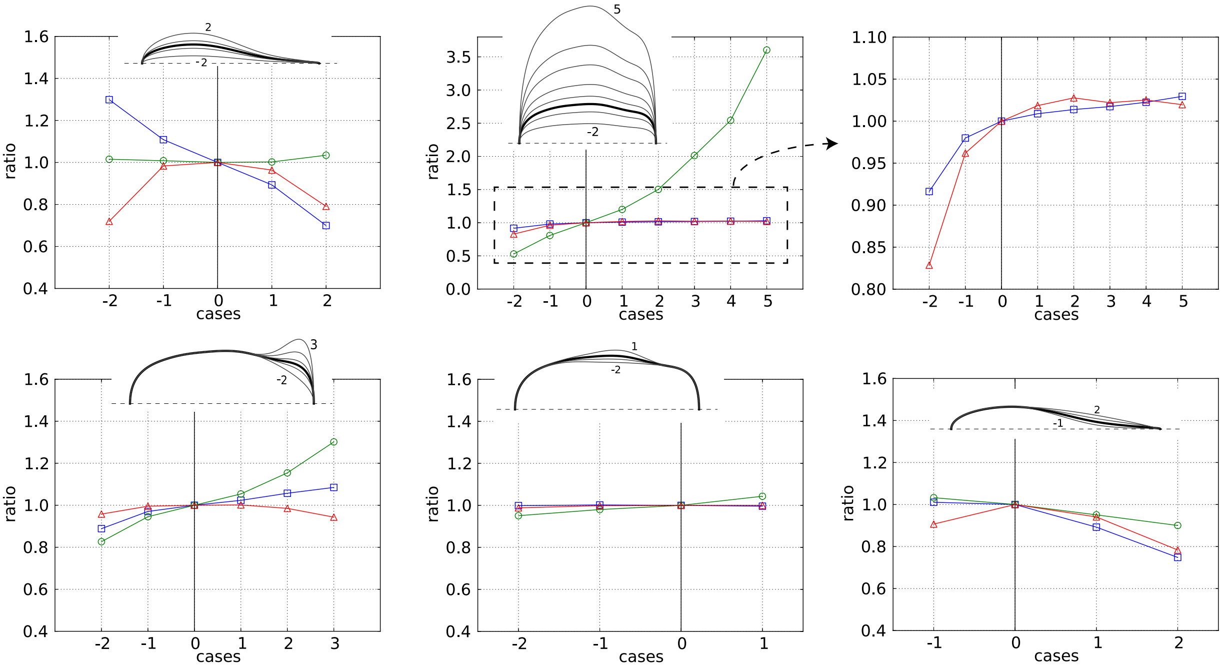


Figure 7.8: Effect of shape variations on the forward velocity (blue squares), efficiency (red triangles) and input energy (green circles) for the efficient swimmer.

7.5.2 Inflection point

The posterior part of the width profile of the optimal swimmers is characterized by an inflection point. The slenderness of the fast swimmer does not allow for a meaningful variation study, and so we focus our analysis here on the efficient case. We chose to enhance the inflection point by decreasing the value of β_3^w (case -1), or to remove it by replacing the profile with a symmetric 4-digit NACA airfoil (Jacobs, Ward, and Pinkerton 1933) with the same maximum thickness. In case 2 we employ this NACA profile, while in case 1 the profile is given by an average between the NACA profile and our optimal solution.

The lack of inflection point is found to be detrimental for both forward velocity (−25%) and efficiency (−20%). We argue, based on the FTLE analysis, that the presence of an inflection point increases the volume of the flow regions trapped by the body and pushed backwards during each stroke, similar to the C-start mechanism (Gazzola, van Rees, and Koumoutsakos 2012). The larger momentum transfer increases the speed, while the associated growth in useful energy dominates the increase in input energy to improve the swimmers' efficiency.

7.5.3 Caudal fin

For the fast swimmer, the caudal fin provides a significant part of the propulsion as shown by the drop in forward velocity when decreasing its size. The presence of a caudal fin increases the available lateral area where lateral velocities are maximal, resulting in a larger momentum transfer and therefore increased acceleration as the swimmer sweeps its tail. However, without a caudal fin the swimmer surprisingly still reaches 80% of its top speed. This shows that the thrust is generated along the entire body, as coherent flow regions are pushed backwards and imparted a final acceleration by the caudal fin. Increasing the caudal fin size beyond the optimum in turn decreases the swimmer's speed. In particular, a larger caudal fin increases frictional resistance and flow separation due to a stronger adverse pressure gradient downstream of the caudal peduncle.

For the efficient swimmer, as for the fastest one, a larger caudal fin improves the forward velocity and thereby increases its useful energy. However, the larger volume of fluid displaced requires more input energy, which dominates the growth of useful energy. Therefore swimmers with largest caudal fins are the least efficient. This trend is also found for the fast swimmer, where the efficiency peaks at case -2.

7.5.4 Height

We find that increasing the fast swimmer's height beyond the bounding box results in a marginal increase in forward velocity, contained within $\sim 4\%$ of case 0. For the efficient swimmer, case 2 (also beyond the bounding box) represents a maximum with $\sim 3\%$ improvement over case 0. In both cases increasing the height leads to a significant increase in the input energy required to sustain swimming, indicating that the highest shapes may not be practically feasible. These considerations justify *a posteriori* the choice of constraining the shape optimization within a bounding box.

7.5.5 Head profile, dorsal fin and caudal peduncle

Naturally occurring anguilliform swimmers are often characterized by a torpedo-like head profile, pronounced dorsal fin and caudal peduncle. Varying these features in our optimal shapes leads to negligible performance degradation (at most $\sim 3\%$) for both speed and efficiency. Therefore, the widespread appearance of these features is not justified for speed or efficiency alone. Such traits may additionally improve maneuverability or serve biological constraints.

7.6 Conclusions

We have presented a reverse engineering process to obtain the fastest and most efficient shapes for self-propelled anguilliform swimmers at moderate Reynolds numbers. This optimization relies on coupling an evolution strategy with three-dimensional flow simulations using remeshed vortex methods.

Flow analysis reveals that the midline kinematics drives the formation of coherent flow regions along either side of the swimmer, that turn pushed downstream by the body traveling wave. Through this mechanism the swimmer transfers momentum to the flow and, consequently, accelerates.

The optimal shapes possess distinct features that are also widespread across natural swimmers, such as a streamlined profile, dorsal fin, caudal fin and caudal peduncle. However, the overall appearance of the optimal morphologies differs from natural anguilliform swimmers. The obtained optimal shapes outperform the larval zebrafish geometry by 40% and 135% for speed and efficiency, respectively. We note that the optimal solutions require large input power (3 to 4 times more than a zebrafish larva), which may not be attainable by natural swimmers due to muscular or metabolic limitations, but might be achievable by robotic devices. The most relevant morphological features identified are the swimmers' thickness, inflection point and caudal fin. Interestingly, the removal of the caudal fin from the fastest shape still allows it to reach 80% of its maximum speed, showing how anguilliform swimming generates thrust along the entire body. Furthermore, based on flow analysis, we argue that the presence of an inflection point allows an increased momentum transfer, improving speed (+25%) and efficiency (+20%).

This study shows that engineered swimmers have the potential to outperform natural anguilliform swimmers in speed and efficiency, for the flow regime considered herein.

Acknowledgments This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s321.

CHAPTER 8

Optimal morphokinematics for undulatory swimmers at intermediate Reynolds numbers

Chapter 7 presented our optimization for swimmers' morphologies given a fixed, anguilliform midline kinematics. We analyzed the steady-state performance gains in speed and efficiency, and the hydrodynamic mechanisms involved. In this chapter we take the approach one step further and perform a combined optimization of morphology and kinematics. Such morphokinematic optima are expected to reflect the intrinsic hydrodynamic interplay between shapes and kinematics. In the following we present our parameterizations, the optimal solutions and an analysis comparing fast, efficient and natural swimmers.

8.1 Introduction

Undulatory swimming is an evolutionary convergent mode for aquatic locomotion at intermediate and high Reynolds numbers. Although the effects of body kinematic properties, such as length, tail beat amplitude and frequency, on hydrodynamic performance are well understood (Gazzola, Argentina, and Mahadevan 2014), the hydrodynamic interplay between the swimmer's morphology and its flexural movements remains elusive.

Observations of natural swimmers suggest biologically favorable morphokinematic combinations since specific gaits are often associated with typical morphological traits (Webb 1984). However, the evolution of swimmer shapes and gaits are not exclusively determined by hydrodynamics, as they are constrained by physiological needs as well as the properties of natural materials. Moreover, natural swimmers are able to actively employ different kinematics according to their needs, e.g. fast bursts during feeding or efficient swimming during migration (Webb 1984; Blake 2004; Kern and Koumoutsakos 2006). Hence, it is likely that artificial swimmers specialized for single hydrodynamic metrics such as speed or efficiency will not converge to naturally existing morphokinematic solutions. Conversely, robotic swimmers have, in principle, the potential to outperform natural swimmers for specific tasks.

Numerical simulations enable direct comparisons between natural and artificial morphokinematic combinations. Computationally inexpensive low-order models of swimming bodies have been coupled to optimization techniques in Tokic and Yue (2012) and Eloy (2013). It was found that the most efficient solutions resemble natural thunniform swimming (Tokic and Yue 2012; Eloy 2013), whereas the fastest solutions depart from existing biological swimmers (Eloy 2013). To obtain a fundamental and detailed insight into the hydrodynamic processes associated with swimming, physically accurate direct numerical simulations are desirable. Such viscous simulations were used in Tytell, Borazjani, et al. (2010) to highlight the intimate interplay between shape and kinematics as quantified in terms of speed and efficiency, and showed that artificial morphokinematic combinations can outperform natural swimmers. More recently, following the work of Kern and Koumoutsakos (2006), direct numerical simulations have been coupled to evolutionary strategies for the identification of optimal escape mechanisms (Gazzola, van Rees, and Koumoutsakos 2012) and morphologies (van Rees, Gazzola, and Koumoutsakos 2013) of larval fish. These studies show that the naturally prevailing C-start escape mechanism is qualitatively optimal (Gazzola, van Rees, and Koumoutsakos 2012), and confirmed that natural morphologies may be improved to increase speed and efficiency during steady anguilliform swimming (van Rees, Gazzola, and Koumoutsakos 2013).

Here, bearing in mind the growing potential of aquatic robotic applications (M. S. Triantafyllou and G. S. Triantafyllou 1995; Griffiths 2003; Ijspeert et al. 2007), we reverse-engineer relations between morphological traits and kinematics to identify optimal solutions in terms of speed and efficiency, thus discounting the unknown biological constraints natural swimmers may be subject to.

8.2 Inverse design of artificial swimmers

We perform two separate optimizations, one for maximum speed and one for maximum efficiency. In both optimizations we fix the swimmers' length L and their undulatory swimming period T. Based on these parameters we define a swimmer Reynolds number $Re = (L^2/T)/\nu = 550$. The corresponding flow regime is typical of larval zebrafish at 5 days post-fertilization (U. K. Müller, van den Boogaart, and van Leeuwen 2008) and allows for direct comparison with previous studies (Gazzola, van Rees, and Koumoutsakos 2012; van Rees, Gazzola, and Koumoutsakos 2013).

The swimmers' midline kinematics are described with a curvature profile $\kappa(s,t)$, defined as an amplitude vector $K(s)$ multiplied by a traveling wave

$$\kappa(s,t) = K(s)\sin\left[2\pi\left(\frac{t}{T} - \tau\frac{s}{L}\right)\right], \tag{8.1}$$

where s is the arclength, t is the time and τ determines the wavelength. The amplitude vector $K(s)$ is a cubic spline interpolating $N_m = 6$ control points distributed at the locations $s/L = (0, 1/20, 1/3, 2/3, 19/20, 1)$ along the midline (Kern and Koumoutsakos 2006). The curvature values at these control points are denoted as β_k^m with $k \in [0, N_m - 1]$, where both endpoint curvatures are fixed to zero. This leads to four free parameters, $\{\beta_k^m\}_{k=1,\dots,4}$, for the curvature that, along with τ, control the swimming gait. The 3D morphology is defined by width and height profiles described by two parameterized B-spline curves, and the 3D volume is constructed with elliptical cross-sections (van Rees, Gazzola, and Koumoutsakos 2013). The resulting shapes are doubly symmetric and can have varying volumes. As in van Rees, Gazzola, and Koumoutsakos (2013), we use four free parameters for the width profile, $\{\beta_i^w\}_{i=1,\dots,4}$, and six free parameters for the height profile, $\{\beta_j^h\}_{j=1,\dots,6}$, resulting in a total of ten parameters determining the morphology. The morphokinematic optimization problem therefore entails a total of fifteen degrees of freedom.

The simulations of self-propelled swimmers in an incompressible, viscous flow are performed with a remeshed vortex method combined with penalization and projection techniques (Gazzola, Chatelain, et al. 2011). The fluid-structure interaction solver is coupled with the stochastic optimization algorithm, the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) in its multi-host, rank-μ and weighted recombination form (Hansen, S. D. Müller, and Koumoutsakos 2003). As in van Rees, Gazzola, and Koumoutsakos (2013), speed is evaluated by averaging the swimmer's forward velocity in the sixth swimming cycle, which we consider representative of steady swimming after the transient starting acceleration. Efficiency is defined as the ratio of useful kinetic energy over the sum of useful energy and energy delivered to the flow, again averaged over the sixth swimming cycle. Since the useful energy contains the product of the swimmer's volume and velocity, this definition may be understood as a transport efficiency.

We constrain the search space for kinematics and morphology to prevent unattainable solutions. For the midline kinematics the absolute curvature of each control point remains smaller than $3\pi/L$. For the morphological parameters we impose that the resulting shapes fit within a bounding box of size $L \times 0.6L \times 0.6L$, as in van Rees, Gazzola, and Koumoutsakos (2013). Unfeasible morphokinematic solutions involving profile curves intersecting each

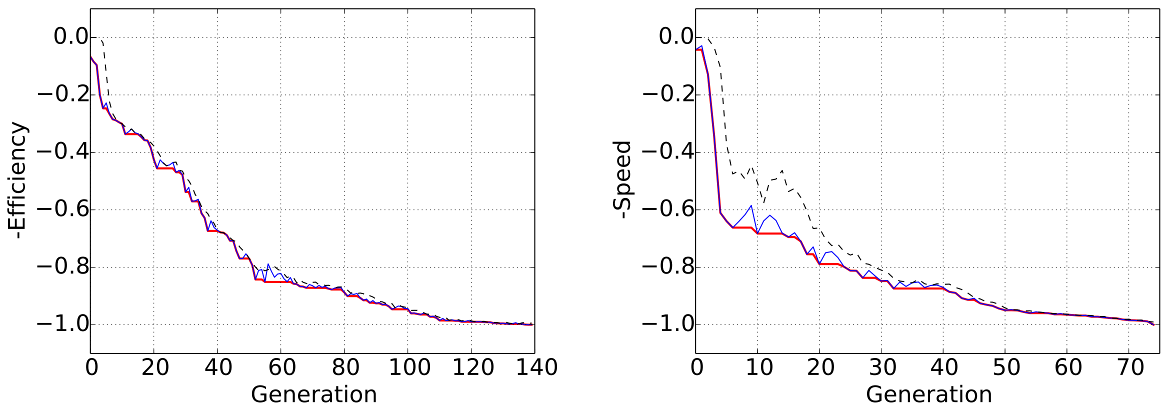


Figure 8.1: The fitness evolution for the efficient (left) and fast (right) swimmer during the optimization. We show the mean per generation (black), the minimum per generation (blue) and the global minimum (green). The y-axes are normalized by their respective optimum values.

	β_1^m	β_2^m	β_3^m	β_4^m	τ
$\mathbf{m}_0$	0	0	0	0	π
$\mathbf{m}_{\text{eff}}$	−2.9	2.0	−6.1	−2.5	2.4
$\mathbf{m}_{\text{fast}}$	2.5	−7.7	−2.2	−2.0	0.76

	β_1^w	β_2^w	β_3^w	β_4^w
$\mathbf{m}_0$	$8.9e^{-2}$	$1.7e^{-2}$	$1.6e^{-2}$	$1.3e^{-2}$
$\mathbf{m}_{\text{eff}}$	$2.4e^{-1}$	$3.0e^{-1}$	$8.3e^{-3}$	$5.7e^{-3}$
$\mathbf{m}_{\text{fast}}$	$1.4e^{-1}$	$8.1e^{-2}$	$5.0e^{-3}$	$7.5e^{-3}$

	β_1^h	β_2^h	β_3^h	β_4^h	β_5^h	β_6^h
$\mathbf{m}_0$	$5.5e^{-2}$	$6.8e^{-2}$	$7.6e^{-2}$	$6.4e^{-2}$	$7.2e^{-3}$	$1.1e^{-1}$
$\mathbf{m}_{\text{eff}}$	$2.2e^{-1}$	$2.7e^{-1}$	$2.8e^{-1}$	$3.0e^{-1}$	$3.0e^{-1}$	$2.2e^{-1}$
$\mathbf{m}_{\text{fast}}$	$6.0e^{-2}$	$4.5e^{-2}$	$8.5e^{-2}$	$9.8e^{-2}$	$9.5e^{-2}$	$1.3e^{-1}$

Table 8.1: The optimal solutions' motion (top left), width (top right) and height (bottom) parameters for fast ($\mathbf{m}_{\text{fast}}$) and efficient ($\mathbf{m}_{\text{eff}}$) swimming, as well as for start searching point $\mathbf{m}_0$. The parameters correspond to swimmers with $L = 1$.

other or the body midline are rejected. This approach prevents the generation of irrelevant engineering as well as numerically unstable solutions, while retaining its generality given the richness of representable morphokinematic solutions. Further details are discussed in the *Supporting Information* (SI).

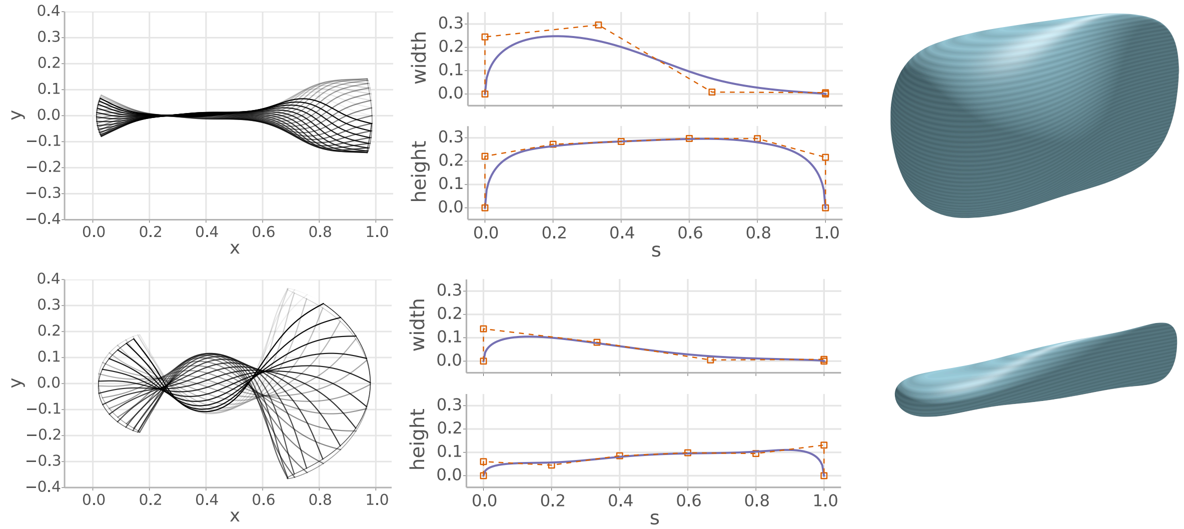


Figure 8.2: Optimal solutions for efficient (top) and fast (bottom) swimmer. The leftmost column shows the midline deformation during one cycle, the middle column contains the width profile (top) and height profile (bottom) for each swimmer and the right column shows a 3D rendering of the optimal shapes.

8.3 Optimal morphokinematics of artificial swimmers

We optimize the morphokinematics of self-propelled swimmers with respect to the efficiency and speed metrics defined above. The optimal solutions (figure 8.2 and table 8.1) were obtained after 140 generations for the efficient metric and 75 for the fast metric, corresponding to 8400 and 4500 individual simulations respectively (figure 8.1). In this section we will discuss the morphology and kinematics of the optimal solutions, subsequently the next section will detail the corresponding swimming mechanisms from a hydrodynamical perspective.

The most efficient swimmer has midline kinematics that remind of carangiform swimming. Lateral deformations are mostly confined to the tail, combined with smaller lateral excursions at the head and almost no deformations in the middle portion of the body. The shape combines a nearly constant maximal height profile with a large width in the anterior part, sharply tapering off towards the tail via a pronounced inflection point to form a thin caudal fin.

The geometry of the fastest swimmer is characterized by a slender and narrow profile, with the height increasing and the width decreasing from the head to the tail. The anterior features are characteristic of a streamlined, low pressure-drag body whereas the taller and thin tail resembles that of a natural swimmer in the absence of the caudal peduncle. The

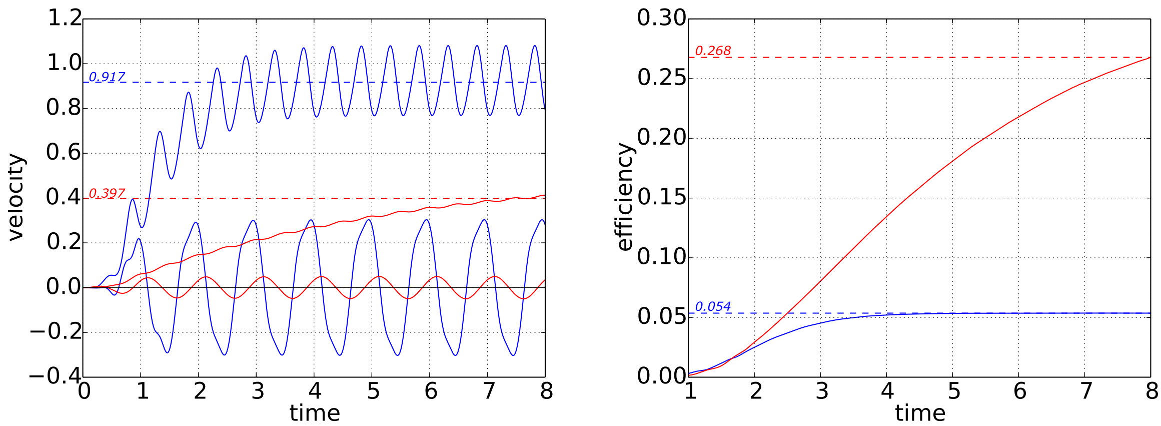


Figure 8.3: The evolution of forward and lateral velocities (left) and efficiency in the last cycle (right) for the fast (blue) and efficient (red) swimmer. The final values are given by the respective numbers above the dashed lines. The numerical data corresponds to swimmers with $L = 1$, $T = 1$ and $Re_{\text{fish}} = 550$.

midline kinematics of the fast swimmer resemble that of the C-start escape mechanism. In a previous optimization study the C-start was shown to be an optimal escape pattern for zebrafish larvae in this Reynolds number regime (Gazzola, van Rees, and Koumoutsakos 2012). The present results suggest that such a gait is optimal also for fast steady swimming, providing an acceleration burst during each cycle that sustains a high average forward velocity (figure 8.3).

8.4 Hydrodynamics of optimal artificial swimmers

Both swimmers create lateral vortex rings in the wake, with each stroke first closing the upstream half of the previous ring and then, as the tail changes direction, shedding the downstream half of the next ring (figure 8.4). Due to the high speed of the fastest swimmer its vortex rings are strongly elongated and resemble those of the C-start escape (Gazzola, van Rees, and Koumoutsakos 2012). The C-start mechanism relies on a strong body curvature to trap a large volume of fluid, that is accelerated backwards during the subsequent propulsive stroke so that the swimmer accelerates in opposite direction (Gazzola, van Rees, and Koumoutsakos 2012). Each stroke of our optimally fast solution may therefore be seen as generating a burst acceleration, which cyclically repeats to sustain a high averaged velocity.

The vortical structures of the efficient swimmer's wake are much weaker. Due to the swimmer's lower speed and tall tail, elliptical vortex rings are shed that then converge

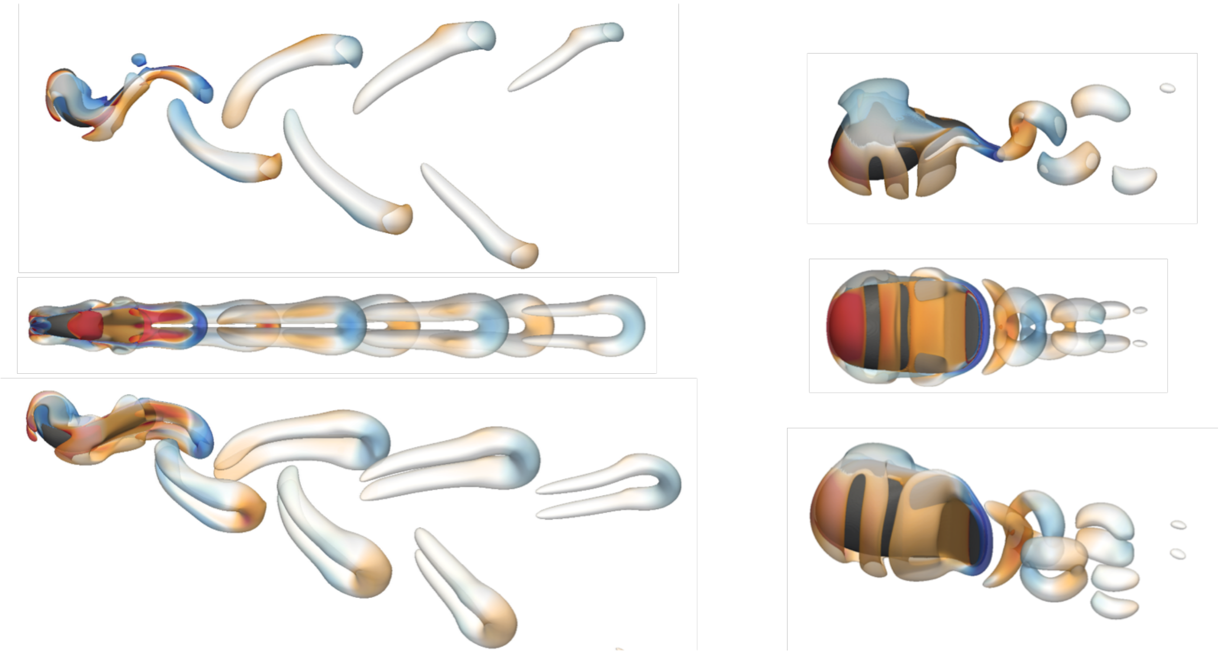


Figure 8.4: Visualization of the vortical structures in the wake using an isosurface of the Q-criterion (Hunt, Wray, and Moin 1988), colored by the local strength of the vertical vorticity component. The swimmers are depicted in dark gray. Top view (top), side view (middle) and perspective view (bottom) of the fast (left) and efficient swimmer (right).

to an almost circular configuration before their energy dissipates. The structures are qualitatively similar to those of the most efficient morphology for anguilliform swimming identified in van Rees, Gazzola, and Koumoutsakos (2013). That morphology shares with our most efficient solution a pronounced inflection point in the width profile. From a hydrodynamical perspective, this is consistent with the argument that steady-state undulatory swimming relies on the same fundamental mechanism as the C-start (van Rees, Gazzola, and Koumoutsakos 2013). In particular, the fluid is trapped inside convex regions on either side of the midline created by the undulatory body deformation, and accelerated backwards by the traveling wave. The inflection point located about midway in both solutions' width profiles favors this mechanism, by increasing the volume of fluid that can be accommodated and accelerated backwards.

8.5 Navigating the morphokinematic landscape

To investigate the interplay between gait and morphology, we explore the morphokinematic landscape between the optimal solutions, and compare with biological swimmers. We

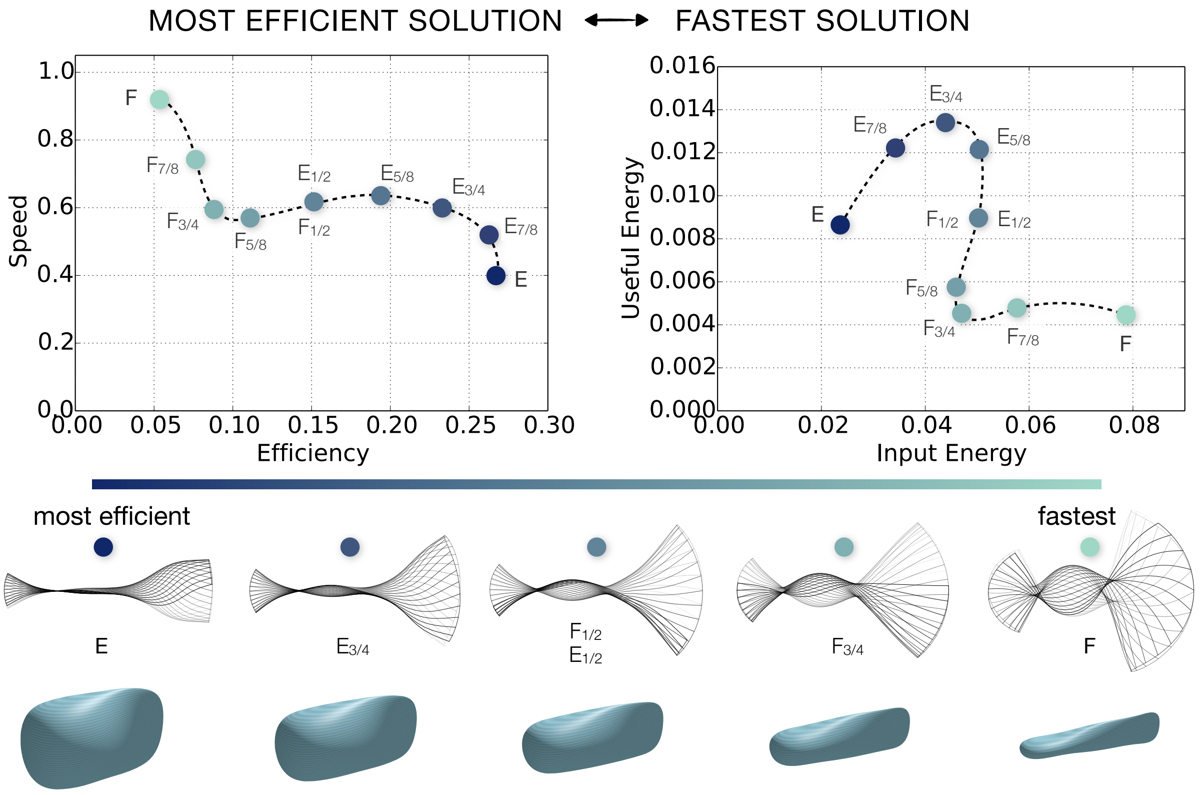


Figure 8.5: Results of the most efficient (E), fastest (F), and seven intermediate solutions obtained by equidistant linear combinations of the fast and efficient swimmers' parameters. The shapes and gaits of the two optima and three intermediate solutions are provided under the plots for reference.

quantify hydrodynamic performance in terms of speed and efficiency as well as useful and required energy.

8.5.1 Transition between optimal solutions

To examine morphokinematic solutions that lie in between the fastest and most efficient solutions identified here, we simulated several swimmers constructed by interpolating all the optimization parameters between the two optimal solutions, as depicted in figure 8.5.

Comparing the fastest swimmer (F) with its two closest hybrid solutions ($F_{7/8}$ and $F_{3/4}$) shows how this relatively small change in shape and gait decreases the speed by about 35%, requiring about 40% less energy to swim. The fastest solution therefore occupies a highly specialized place in our parameter landscape, likely requiring too much extra energy with respect to slower solutions to be biologically attainable for steady-state swimming.

On the other side of the parameter space, the second-most efficient solution ($E_{7/8}$) still attains about 80% of the efficiency of the optimal case (E). It reaches this efficiency even though it is about 25% faster, and has about 40% higher useful energy, because the required energy increases correspondingly. These results imply that relatively high efficiencies can be obtained within a range of shape-gait combinations, making efficient solutions less specialized and therefore more robust than fast swimmers.

We note that the five most central swimmers ($E_{3/4}$ to $F_{3/4}$) achieve approximately equal speeds with similar energetic requirements, both quantities lying between those of the fastest and most efficient solutions. This plateau in the morphokinematic landscape provides a space in which swimmers' designs can be specialized for different tasks or environments, without compromising their hydrodynamic performance.

8.5.2 Comparison with anguilliform larval zebrafish

We compare our optimal solutions with simulations of a natural anguilliform larval zebrafish and, as in Borazjani and Sotiropoulos (2010), we interchange morphologies and kinematics to assess the individual contributions of shape and gait. We denote morphokinematic combinations with a two letter sequence, using a subscript to specify morphology or gait. The combination of zebrafish morphology (Z_m) and anguilliform gait (A_g) is thus denoted Z_mA_g, whereas the fastest and most efficient solutions are denoted F_mF_g and E_mE_g, respectively. We will refer to the optimal morphologies of van Rees, Gazzola, and Koumoutsakos (2013), which were optimized with the fixed anguilliform gait of the zebrafish for fast and efficient swimming, as $\tilde{F}_m$ and $\tilde{E}_m$. We discuss separately the cases of efficiency and speed, as presented in Figs. 8.6 and 8.7, respectively.

Efficient swimming For the anguilliform gait, exchanging the zebrafish shape (Z_mA_g) with the optimal shape (E_mA_g) almost triples the efficiency, as the volume is increased twentyfold whereas the steady-state speed remains almost constant. This indicates that the propulsive benefits of the larger lateral area are sufficient to compensate the increased pressure and friction drag forces.

Maintaining the optimal solution's morphology and interchanging the anguilliform gait (E_mA_g) with the optimal gait (E_mE_g) increases the speed by 30% and efficiency by 60%, without significantly altering the energy expenditure of the swimmer. Yet for the zebrafish morphology, replacing the anguilliform gait (Z_mA_g) with the gait of the optimal solution (Z_mE_g) decreases both the speed and the efficiency by about 50%. An identical change in gait thus improves the performance of one morphology and degrades it for another,

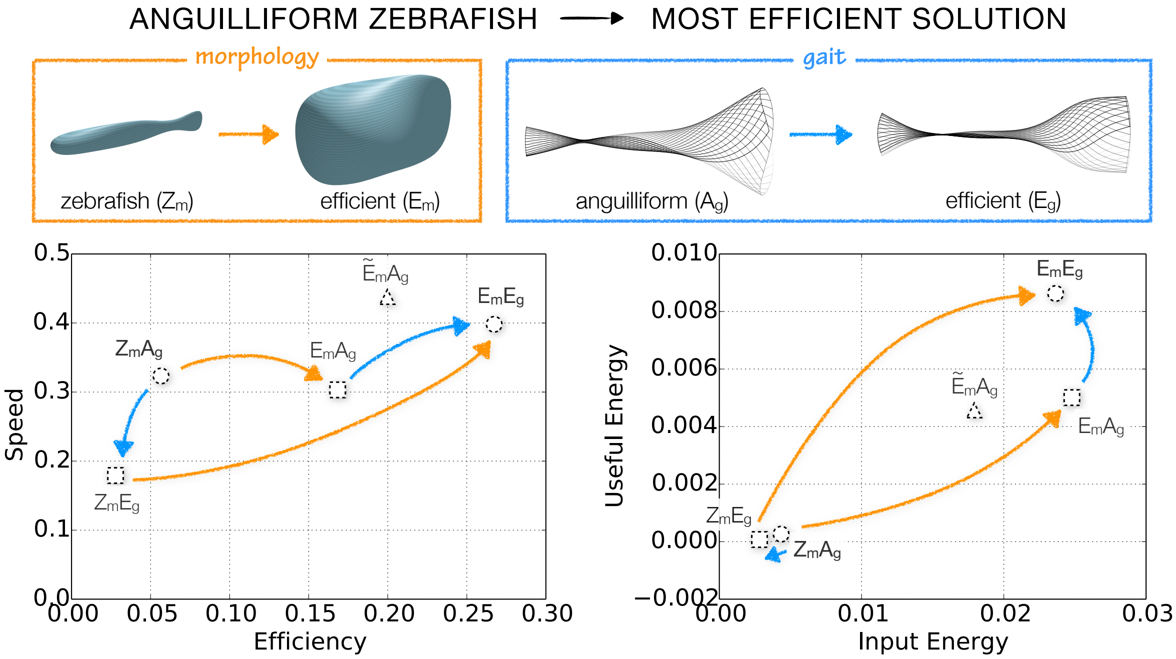


Figure 8.6: Comparing the most efficient swimmer (E_mE_g) with the anguilliform zebrafish (Z_mA_g), and two intermediate solutions interchanging the zebrafish shapes and gaits with those of the optimal solution. The triangle $\tilde{E}_mA_g$) is the optimal shape for efficient anguilliform swimming as found in van Rees, Gazzola, and Koumoutsakos (2013).

stressing the hydrodynamic interplay between morphology and midline kinematics in undulatory swimming.

We note that the anguilliform swimmer with the optimal solution's morphology (E_mA_g) has only slightly lower efficiency than the most efficient anguilliform swimmer ($\tilde{E}_mA_g$), which in turn reaches already 75% of the efficiency of the optimal solution (E_mE_g). Anguilliform midline kinematics seem therefore well suited for high efficiency swimming, provided the morphology is optimized accordingly.

Fast swimming Figure 8.7 shows that when the anguilliform zebrafish (Z_mA_g) adopts the gait of the fastest solution, the resulting swimmer (Z_mF_g) reaches almost 90% of the optimal solution's (F_mF_g) speed. On the other hand, an anguilliform swimmer with the optimal solution's shape (F_mA_g) reaches about the same velocity as the anguilliform zebrafish (Z_mA_g). This indicates that gait rather than morphology is the predominant factor in fast swimming, correlating with the observation that fish generally can actively adjust their gait, whereas only very few can adjust their morphology. The performance increase of the fastest solution's gait comes with an energetic cost, however, as the Z_mF_g

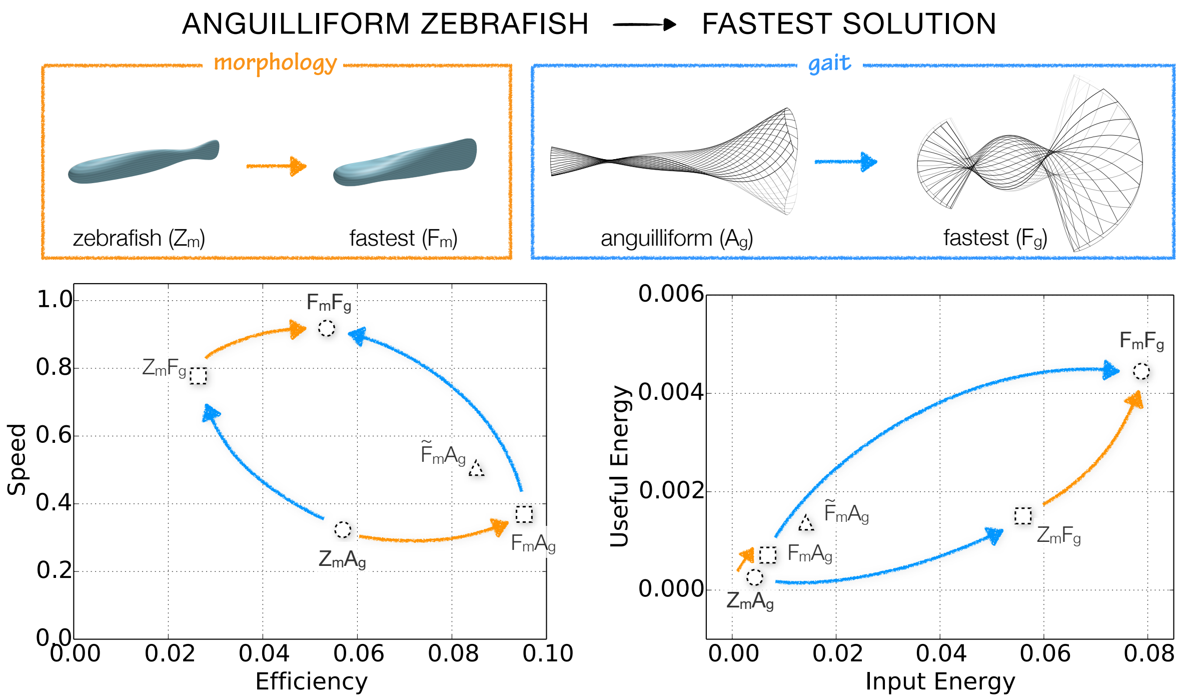


Figure 8.7: Comparing the fastest swimmer (F_mF_g) with the anguilliform zebrafish (Z_mA_g), and two intermediate solutions interchanging the zebrafish shapes and gaits with those of the optimal solution. The triangle ($\tilde{F}_mA_g$) is the optimal shape for fast anguilliform swimming as found in van Rees, Gazzola, and Koumoutsakos (2013).

swimmer requires 13 times more energy and sees a 50% drop in swimming efficiency compared with the anguilliform gait (Z_mA_g). The fastest solution's gait might thus only be viable for natural swimmers during short periods of time, but could well be considered for more powerful robotic swimmers.

For the anguilliform swimming gait, we note that the shape optimized for fast anguilliform swimming ($\tilde{F}_mA_g$) is only about 40% faster than the anguilliform zebrafish (Z_mA_g), whereas the optimal shape-gait combination (F_mF_g) is almost 300% faster. This implies that the anguilliform kinematics are ill-suited for fast swimming, even with a shape optimized for speed.

8.5.3 Strouhal number for efficient swimming

The reduced swimming frequency, in the context of undulatory swimming commonly denoted by the Strouhal number, is defined as $St = fA/U$, where $f = 1/T$, A is the lateral distance traveled by the tail during one stroke and U is the averaged forward velocity. Natural swimmers typically operate at $St \sim 0.3$, which coincides with the maximum efficiency for thrust-producing airfoils (G. S. Triantafyllou, M. S. Triantafyllou, and Grosenbaugh 1993; Taylor, Nudds, and Thomas 2003). In contrast, our most efficient swimmer has a Strouhal number of 0.72, and a previous optimization study for efficient swimming kinematics found a Strouhal number of 0.67 (Kern and Koumoutsakos 2006).

Following Gazzola, Argentina, and Mahadevan (2014), however, we have $St \sim Re_U^{-1/4}$, where $Re_U = UL/\nu$, so that a tenfold reduction in the Reynolds number will double the Strouhal number. Indeed the Reynolds numbers in this work, as well as in Kern and Koumoutsakos (2006), are more than an order of magnitude smaller than the data in G. S. Triantafyllou, M. S. Triantafyllou, and Grosenbaugh (1993) and Taylor, Nudds, and Thomas (2003), and the corresponding Strouhal numbers are more than a factor of two larger. We therefore conclude that the high Strouhal number of our optimally efficient swimmer is consistent with the principal scaling laws governing undulatory self-propelled swimming.

8.6 Conclusions

We have reverse-engineered undulatory self-propelled swimmers to maximize speed or efficiency as a function of both their morphology and gait. The optimal solutions outperform natural zebrafish by over 4.5 times for efficiency, or 2.5 times for speed. The fastest swimmer uses a succession of C-start swimming strokes characterized by large curvatures along the entire body, combined with a streamlined geometry profile. This result is consistent with previous studies, but may not be energetically viable for biological swimmers due to the extremely high energetic requirements. The most efficient swimmer is characterized by a large volume and carangiform-like midline kinematics, where the largest curvatures are confined to the tail and, to lesser extent, the head.

We investigated the morphokinematic landscape by examining swimmers obtained from interpolating the shape-gait parameters of the fastest and the most efficient solutions. The results show that the speed of the fastest solution is very sensitive to parameter changes, making it a highly specialized, non-robust solution. Such fast swimming requires a high input energy, consistent with previous results. In contrast the efficient swimmer is more robust to parameter deviations, as large fractions of its value can be obtained even by hybrid solutions. In between we discovered a range of swimmers with different morphokinematic characteristics but similar speeds and energetic requirements. Within this region, specialization dictated by various endogenous and exogenous constraints may take place without compromising hydrodynamic performance.

The optimal solutions identified in the present work were compared with natural anguilliform zebrafish, as well as optimal shapes for anguilliform kinematics (van Rees, Gazzola, and Koumoutsakos 2013). We showed that an anguilliform swimming gait can support highly efficient swimmers, as long as the shape is suitably adapted. The actual efficiency value depends stronger on the shape than on the gait, in the context of our definition of transport efficiency. In contrast, fast swimming relies more on gait than on the shape, and indeed natural zebrafish larva geometries can reach 90% of the fastest morphokinematic solution's speed just by adopting the corresponding C-start kinematics. We conclude that a natural swimmer could reach an impressively fast speed by suitably altering its midline kinematics, but note that the energy required to sustain this speed is up to 20 times larger than for anguilliform swimming.

Our results indicate that the present inverse-design framework is capable of identifying self-propelled swimmers that outperform biological solutions. By analyzing the features of the optimal swimmers, and exploring their morphokinematic landscape, this work provides a compass for the design of high performance artificial swimmers for engineering applications.

Acknowledgments This work was performed together with Dr. Mattia Gazzola (Harvard University), and was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s435.

Part III

Multiresolution flow simulations

CHAPTER 9

Space-time adaptive methods

High Reynolds number flows such as the vortex reconnection presented in part I are often characterized by a strong disparity in spatiotemporal scales. Similarly, flows with fluid-solid interaction such as shown in part II combine small-scale boundary layers around the body with interacting vortices in the wake and a quiescent flow upstream of the body. The efficient simulation of such flows requires the adaptation of the computational elements to the governing structures of the flow field.

The remeshed vortex method as presented in section 2.3 already provides an intrinsic adaptation of computational elements by exploiting the compact support of the vorticity field. A further efficiency gain can be achieved by using computational elements that adapt to the spatiotemporal scales in the flow.

Currently, most adaptive-grid flow simulations use either Adaptive Mesh Refinement (AMR) (Almgren et al. 1998; Martin and Colella 2000) or wavelet-based multiresolution techniques (Schneider and Farge 2002; Vasilyev and Kevlahan 2002). The advantages of wavelet-based methods are discussed in Cohen (2002) and their use in fluid mechanics were reviewed in Schneider and Vasilyev (2010). Wavelets provide a consistent multiresolution framework which leads to higher data compressions, explicit error control and consistent reconstruction of ghosts. A challenge in their use is the efficient implementation on multi/many-core architectures. In this chapter we further discuss these issues and present our contributions.

In section 9.1 we will discuss briefly the mathematical details of the multiresolution framework provided by wavelets, with specific focus on the interpolating wavelets used in this thesis. Then we describe in section 9.2 how wavelets can be used for the solution of partial differential equations (PDEs) with grid-based and particle-based methods. In section 9.3 we show our implementation of a 3D wavelet-based multiresolution rVM and the optimizations for multicore architectures. The results of adaptive-grid flow simulations will be presented in the next two chapters.

9.1 Wavelet-based multiresolution analysis

We summarize the mathematical details of wavelet-based multiresolution analysis that are relevant to this work. We start with the framework of multiresolution analysis, and how this fits with orthogonal wavelets and its extension to biorthogonal wavelets. Then we discuss the interpolating wavelets used in our multiresolution simulations, and their extension to finite-sized domains.

9.1.1 Multiresolution analysis

The multiresolution analysis (MRA) (Mallat 1989) decomposes the space $L_2(\mathbb{R})$ into a sequence of nested subspaces $\left(V_j\right)_{j\in\mathbb{Z}}$

$$0\cdots\subset V_{-1}\subset V_0\subset V_1\subset\cdots\subset L_2(\mathbb{R}), \tag{9.1}$$

so that the union $\bigcup_{j\in\mathbb{Z}} V_j$ is dense and the intersection $\bigcap_{j\in\mathbb{Z}} V_j$ is empty. The spaces satisfy the dilatation property $f(x)\in V_j \iff f(2x)\in V_{j+1}$, and the reference space V_0 is spanned by an orthonormal basis consisting of integer translates of a basis function $\phi(x)$, namely $\{\phi(x-k), k\in\mathbb{Z}\}$. The basis function $\phi(x)$ is called the scaling function, or father wavelet, and it satisfies the orthonormality condition $\langle\phi(x),\phi(x-k)\rangle=\delta_{0,k}$. Using the notation

$$\phi_{j,k}(x)=2^{j/2}\phi(2^j x-k), \tag{9.2}$$

to represent dilatations and translations of a function, we have that each space V_j is spanned by $\phi_{j,k}(x)=\sqrt{2}\phi_{j-1,k}(2x), k\in\mathbb{Z}$.

9.1.2 Orthogonal wavelets

The difference between two spaces V_{j-1} and V_j is characterized by the introduction of a new subspace W_j as the orthogonal complement of V_j to V_{j+1}, so that $V_j\perp W_j$ and $V_{j+1}=V_j\oplus W_j$. Together with subspace W_j a new basis function $\psi(x)$ is introduced so that $\psi_{j,k}(x)$ forms an orthonormal basis for W_j. Choosing an arbitrary level $j=0$, the multiresolution analysis guarantees that we can describe any function in V_0 and W_0 using the basis functions of V_1, so that

$$\phi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} h_k \phi(2x-k), \tag{9.3}$$

$$\psi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} g_k \phi(2x-k), \tag{9.4}$$

where the filter coefficients are $h_k = \left\langle \phi(x), \sqrt{2}\phi(2x-k)\right\rangle$ and $g_k = \left\langle \psi(x), \sqrt{2}\phi(2x-k)\right\rangle$. Since W_0 is the orthogonal complement of V_0, we have $\langle \psi(x), \phi(x)\rangle = 0$, which gives

$$g_k = (-1)^k h_{-k+1}. \tag{9.5}$$

The function $\psi(x)$ is called the wavelet function, or mother wavelet. Classical examples of scaling functions $\phi(x)$ are the Haar wavelet and the Daubechies wavelet family. The latter is constructed by solving for the coefficients h_k given the restriction that the basis $\{\phi(x-k), k\in\mathbb{Z}\}$ is orthonormal and can exactly reproduce polynomial functions up to a specified order. The coefficients h_k are then substituted into equation (9.3) and $\phi(x)$ is solved from the resulting implicit set of equations.

Using these orthogonal wavelets one can project any function $f(x)$ onto a space V_j using

$$f^j(x) = \sum_{k\in\mathbb{Z}} \left\langle f, \phi_{j,k}\right\rangle \phi_{j,k}. \tag{9.6}$$

The difference between the projections f^{j+1} and f^j can be expressed in terms of $\psi(x)$

$$\sum_{k\in\mathbb{Z}} \left\langle f, \phi_{j+1,k}\right\rangle \phi_{j+1,k} = \sum_{k\in\mathbb{Z}} \left\langle f, \phi_{j,k}\right\rangle \phi_{j,k} + \sum_{k\in\mathbb{Z}} \left\langle f, \psi_{j,k}\right\rangle \psi_{j,k}. \tag{9.7}$$

We can apply this recursively to decompose any function $f(x)\in L_2(\mathbb{R})$ into its representation in a reference space V_J and its orthogonal complements $W_J \oplus W_{J+1} \oplus \cdots$, so that

$$f(x) = \sum_{k\in\mathbb{Z}} c_k^J \phi_{J,k} + \sum_{j=J}^{\infty}\sum_{k\in\mathbb{Z}} d_k^j \psi_{j,k}(x), \tag{9.8}$$

where

$$c_k^j = \left\langle f(x), \phi_{j,k}(x))\right\rangle, \tag{9.9}$$

$$d_k^j = \left\langle f(x), \psi_{j,k}(x)\right\rangle, \tag{9.10}$$

are called the scaling and detail coefficients, respectively.

9.1.3 Fast Wavelet Transform

Combining equations (9.3) and (9.4), with equations (9.9) and (9.10), one can find the following recursive relationships

$$c_k^j = \sum_n h_{n-2k} c_n^{j+1} \tag{9.11}$$

$$d_k^j = \sum_n g_{n-2k} c_n^{j+1}. \tag{9.12}$$

These equations represent the Fast Wavelet Transform (FWT) and constitute a convolution with the filter coefficients, and a subsequent downsampling so that only even entries are retained. For a given wavelet, where h_k and g_k are known, the FWT can be used to efficiently decompose the signal (analysis) into its scaling and wavelet coefficients. For the inverse FWT, one recombines (synthesis) the scaling and detail coefficients of one level to get the scaling coefficients of the next level

$$c_k^{j+1} = \sum_n \left(h_{k-2n} c_n^j + g_{k-2n} d_n^j \right). \tag{9.13}$$

Here the structure for each of the terms is an upsampling of the wavelet coefficients, followed by a convolution with the filter coefficients.

9.1.4 Biorthogonal wavelets

Biorthogonal wavelet bases (Cohen, Daubechies, and Feauveau 1992) relax the criteria for finding scaling and the corresponding wavelet functions, and allow, for instance, the use of symmetric functions as wavelets. With biorthogonal wavelets the basis $\{\phi(x-k), k \in \mathbb{Z}\}$ is no longer orthogonal. Instead one finds a sequence of dual spaces, $\widetilde{V}_j$, of which the reference frame $\widetilde{V}_0$ is spanned by a basis $\{\widetilde{\phi}(x-k), k \in \mathbb{Z}\}$, so that $\left\langle \phi(x), \widetilde{\phi}(x-k) \right\rangle = \delta_{0,k}$. The spaces V_j and $\widetilde{V}_j$ have non-orthogonal complements W_j and $\widetilde{W_j}$, respectively, so that $V_j \perp \widetilde{W_j}$ and $\widetilde{V}_j \perp W_j$. This provides four functions $\phi(x)$, $\widetilde{\phi}(x)$, $\psi(x)$, $\widetilde{\psi}(x)$, that are related as

$$\phi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi(2x-k), \tag{9.14}$$

$$\widetilde{\phi}(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} \widetilde{h}_k \phi(2x-k), \tag{9.15}$$

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \phi(2x-k), \tag{9.16}$$

$$\widetilde{\psi}(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} \widetilde{g}_k \widetilde{\phi}(2x-k). \tag{9.17}$$

Biorthogonality requires that

$$g_k = (-1)^k \widetilde{h}_{-k+1}, \qquad \widetilde{g}_k = (-1)^k h_{-k+1}. \tag{9.18}$$

Now any function $f(x) \in L_2(\mathbb{R})$ can be decomposed as

$$f(x) = \sum_{k\in\mathbb{Z}} \widetilde{c}_k^J \phi_{J,k}(x) + \sum_{j=J}^{\infty} \sum_{k\in\mathbb{Z}} \widetilde{d}_k^j \psi_{j,k}(x) \tag{9.19}$$

$$= \sum_{k\in\mathbb{Z}} c_k^J \widetilde{\phi}_{J,k}(x) + \sum_{j=J}^{\infty} \sum_{k\in\mathbb{Z}} d_k^j \widetilde{\psi}_{j,k}(x), \tag{9.20}$$

where

$$\widetilde{c}_k^j = \left\langle f(x), \widetilde{\phi}_{j,k} \right\rangle, \qquad \widetilde{d}_k^j = \left\langle f(x), \widetilde{\psi}_{j,k}(x) \right\rangle, \tag{9.21}$$

$$c_k^j = \left\langle f(x), \phi_{j,k} \right\rangle, \qquad d_k^j = \left\langle f(x), \psi_{j,k}(x) \right\rangle. \tag{9.22}$$

In the FWT for biorthogonal wavelets the filters for analysis and synthesis are different between the forward and inverse directions. If the forward transform is defined as

$$c_k^j = \sum_n h_{n-2k} c_n^{j+1} \tag{9.23}$$

$$d_k^j = \sum_n g_{n-2k} c_n^{j+1}, \tag{9.24}$$

then the inverse transform is defined as

$$c_k^{j+1} = \sum_n \left(\widetilde{h}_{k-2n} c_n^j + \widetilde{g}_{k-2n} d_n^j \right). \tag{9.25}$$

9.1.5 Interpolating wavelets

Interpolating wavelets are obtained by combining the above multiresolution analysis for biorthogonal wavelets with the multiscale interpolation technique of Dubuc (1986) and Deslauriers and Dubuc (1989). In this interpolation technique, the starting point is a set of function values evaluated at the integers ($v_k = f(k) : k \in \mathbb{Z}$). These function values are used to estimate $f(k+1/2), k \in \mathbb{Z}$ by polynomial interpolation of a specified degree. At the next level one uses those interpolated values together with the original data points to compute $f(k+1/4), k \in \mathbb{Z}$ in a similar manner. Applying this recursively over the real space gives a function approximation $\widetilde{f}(x) \approx f(x)$ that satisfies $\widetilde{f}(k) = v_k$ for $k \in \mathbb{Z}$. By interpolating a data set $v_k = \delta_{0,k}$ with third-degree polynomials one obtains the kernel function $\phi_{(3)}(x)$ illustrated in figure 9.1.

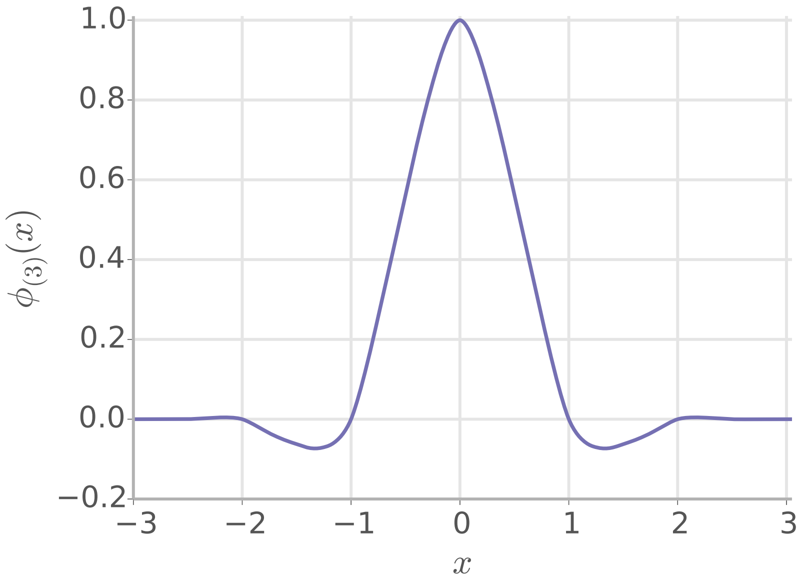


Figure 9.1: Interpolation function $\phi_{(3)}(x)$ obtained through successive refinements of the Deslauriers-Dubuc scheme with third-degree polynomials on the data points $f(k) = \delta_{0,k}, k \in \mathbb{Z}$.

The filter coefficients for this fourth-order interpolating wavelet are given in table 9.1. Using these coefficients, a one-level forward wavelet transform of a signal sequence from fine level $j+1$ to the coarser level j can be interpreted as follows. Setting $c_k^{j+1} = f_k$, where f_k is the value of our sequence at index k, we find for c_k^j

$$c_k^j = \sum_n h_{n-2k} c_n^{j+1} = c_{2k}^{j+1}. \tag{9.26}$$

The scaling coefficients of the coarser representation at level j are just the scaling coefficients associated to the fine grid points with even indices. The detail coefficients are

$$d_k^j = \sum_n g_{n-2k} c_n^{j+1} = c_{2k-1}^{j+1} - \left(-\frac{1}{16} c_{2k-4}^{j+1} + \frac{9}{16} c_{2k-2}^{j+1} + \frac{9}{16} c_{2k}^{j+1} - \frac{1}{16} c_{2k+2}^{j+1} \right). \tag{9.27}$$

The part of this expression between parentheses is a third-degree polynomial constructed from the fine grid points with even indices, evaluated at the center index $2k-1$. The detail coefficients therefore represent the difference between the actual value of c_{2k-1}^{j+1} and its approximation through a third-degree polynomial.

k	-4	-3	-2	-1	0	1	2	3
h_k					1			
g_k	$1/16$	0	$-9/16$	1	$-9/16$	0	$1/16$	
$\widetilde{h}_k$	0	$-1/16$	0	$9/16$	1	$9/16$	0	$-1/16$
$\widetilde{g}_k$					1			

Table 9.1: Filter coefficients for fourth-order interpolating wavelets.

Analogously, the inverse wavelet transform from level j to $j+1$ becomes

$$c_k^{j+1} = \sum_n \left(\widetilde{h}_{k-2n} c_n^j + \widetilde{g}_{k-2n} d_n^j \right) \tag{9.28}$$

$$= \begin{cases} c_{k/2}^j & k \text{ even} \\ d_{(k+1)/2}^j + \left(-\frac{1}{16} c_{(k-3)/2}^j + \frac{9}{16} c_{(k-1)/2}^j + \frac{9}{16} c_{(k+1)/2}^j - \frac{1}{16} c_{(k+3)/2}^j \right) & k \text{ odd} \end{cases} \tag{9.29}$$

which represents the complete reconstruction of the signal at level $j+1$ from its scaling and detail coefficients at level j.

9.1.6 Interpolating wavelets on the interval

Wavelet transforms are usually performed assuming infinite or periodic domains. In our case, periodic boundary conditions are invalid since the vorticity field is compact and the velocity field is solved under the assumption of free-space boundary conditions, so that the velocity and possibly vorticity field will be discontinuous across the domain boundaries. Therefore we extend our 3D wavelet-based multiresolution implementation by implementing interpolating wavelets on the interval (Donoho 1992). These wavelets retain the interpolating advantages described in the previous section, but use one-sided interpolation near domain boundaries. Their stencil and graphical forms are shown in figure 9.2.

9.2 Wavelet-based multiresolution and PDEs

With the mathematical details of the wavelet analysis discussed above, we show here how this multiresolution framework can be applied to the solution of partial differential equations (PDEs). We discuss wavelet-based adapted grids, temporal adaptivity using local time stepping and Lagrangian multiresolution particle methods. The main ideas and successful implementations of this approach have been presented in Hejazialhosseini,

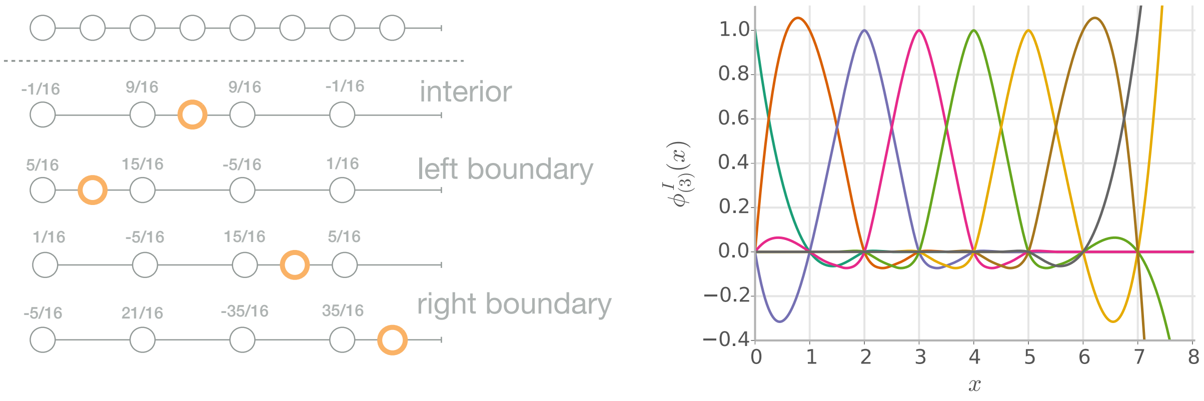


Figure 9.2: On the left, an illustration of the four different interpolating wavelet kernels for computing the detail coefficients of a one-level FWT on a finite 1D grid with eight vertices. The numbers are the weights for the respective coarse points to compute the detail coefficient corresponding to the orange fine point. On the right a plot of the corresponding kernels obtained by successively interpolating the functions $f(k) = \delta_{0,k}$ with k representing the integers in $[0, 8)$.

Rossinelli, Bergdorf, et al. (2010), Rossinelli, Hejazialhosseini, Bergdorf, et al. (2011), and Hejazialhosseini, Rossinelli, Conti, et al. (2012). This section is partially based on Rossinelli, Hejazialhosseini, van Rees, et al. (2015).

9.2.1 Adapted grids

The multiresolution analysis detailed above can be used to compress a data signal by thresholding the detail coefficients that carry negligible information

$$f_\epsilon(x) = \sum_{k\in\mathbb{Z}} c^J_k \phi_{J,k} + \sum_{j=J}^{\infty} \sum_{k\in\mathbb{Z}:|d^j_k|\geq\epsilon} d^j_k \psi_{j,k}(x), \tag{9.30}$$

where ϵ is a user-specified threshold. This introduces an L_∞ error in the signal f that is bounded by ϵ.

Since the scaling and detail coefficients are associated with the indices of a computational grid, we can compress the grid accordingly. Specifically, at the beginning of every N_{adapt} time steps, the flow field is analyzed using a one-level forward wavelet transform and the detail coefficients are compared with a user-specified refinement threshold ϵ_R. If the detail of any relevant quantity exceeds the threshold, we refine the grid to capture the new scales in the flow. Similarly, at the end of every N_{adapt} time steps, we compress if detail coefficients are smaller than a user-specified compression threshold ϵ_C. The

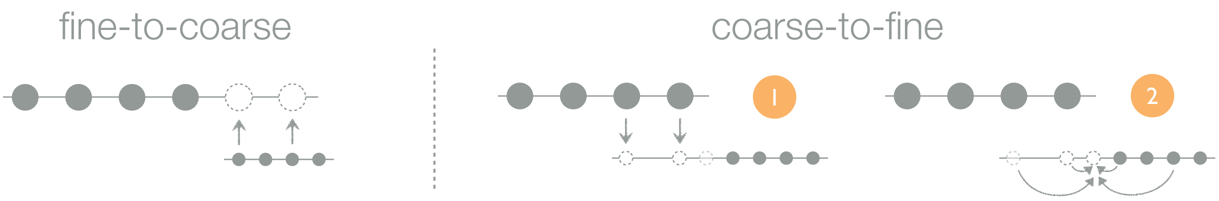


Figure 9.3: Two-point ghost reconstruction for fourth-order interpolating wavelets between two grids that are one level of resolution apart. The coarse-to-fine ghost reconstruction consists of two consecutive stages, as an intermediate third ghost point is required to compute the fine-grid ghosts.

actual compression and refinement stages use forward and inverse wavelet transforms, respectively, to compute the field values at the newly created grid locations.

In order to be able to compute finite-difference (FD) stencils on the adapted grid, we create a frame of uniform resolution around a grid point. This frame is obtained by reconstructing ghost values, obtained as linear combinations of the surrounding active scaling coefficients. The specific weights and source points of a ghost depend on the chosen wavelets as well as the local structure of the adapted grid (Hejazialhosseini, Rossinelli, Bergdorf, et al. 2010; Rossinelli, Hejazialhosseini, Bergdorf, et al. 2011). In figure 9.3 we sketch the process for fourth-order interpolating wavelets with a one-level jump in resolution. After obtaining the ghost points, first and second derivatives of the flow field can be constructed with standard finite-difference techniques.

9.2.2 Local time stepping

Spatially-adapted grids decrease the amount of operations by removing computational elements. Operations can be further decreased by employing local time-stepping (LTS) schemes, which have been shown to accelerate the time-to-solution by over one order of magnitude (Roussel et al. 2003; Alam, Kevlahan, and Vasilyev 2006; Domingues et al. 2008; Domingues et al. 2009; Rossinelli, Hejazialhosseini, Bergdorf, et al. 2011). LTS schemes exploit the spatial locality of the time-stepping stability condition. Grid points are grouped according to their time-stepping constraint and the global time step is chosen according to the coarsest grid spacing. Within a global time step the LTS scheme advances each group according to its own stability condition. If ghost values from coarser resolutions are required at intermediate times they are reconstructed using a temporal interpolation scheme, thus reducing the total number of right-hand side (RHS) evaluations. A second-order accurate total-variation diminishing LTS scheme was presented in Hejazialhosseini, Rossinelli, Bergdorf, et al. (2010) and Rossinelli (2011).

9.2.3 Multiresolution CFL-particles

Hybrid particle-grid methods were first coupled with wavelet-based adaptivity in Bergdorf, Cottet, and Koumoutsakos (2005) and Bergdorf and Koumoutsakos (2006). In this work we consider a simpler multiresolution particle method, employed exclusively for the integration of the convection term. The method leaves the structure of the adapted grid unchanged during the particle-grid operations, and instead relies on the ghost reconstruction discussed above. We first compute a cloud of ghosts around each active scaling coefficient. The cloud size is large enough to guarantee the partition of unity of the particle contributions within the range of one grid spacing. Scaling coefficients and ghosts are then treated as particles and their locations are integrated according to the velocity field. After advection we remesh all the particles, obtaining new values at the initial locations of the scaling coefficients.

When particles are advected with no CFL-like constraint, remeshing can take place across two different levels of resolutions. To avoid this issue we constrain the time step to CFL < 1. For the FSI simulations presented in this part of the thesis, the particle time step does not exceed this threshold without reducing the accuracy of the results, so that the CFL-limitation has no practical impact.

9.3 Implementation on multicore architectures

As noted in the introduction to this thesis, one of the major practical challenges of wavelet-based adaptive-grid simulations are their efficient implementation of modern compute architectures. Here we we discuss our approach to overcome these issues. This section is partially based on Rossinelli, Hejazialhosseini, van Rees, et al. (2015).

9.3.1 Wavelet type

In this work we use fourth-order interpolation wavelets unless otherwise specified. Our multiresolution implementation exclusively relies on first-generation wavelets, whose multiresolution analysis is constructed with dilated and translated versions of just one mother wavelet. Despite the moment-preserving features of second-generation wavelets, first-generation wavelets are computationally advantageous in our approach. Their simpler structure enables the reconstruction pattern for ghost points to be captured and stored into a lookup table for future reuse (Rossinelli, Hejazialhosseini, Bergdorf, et al. 2011). This accelerates the computation of the ghost values.

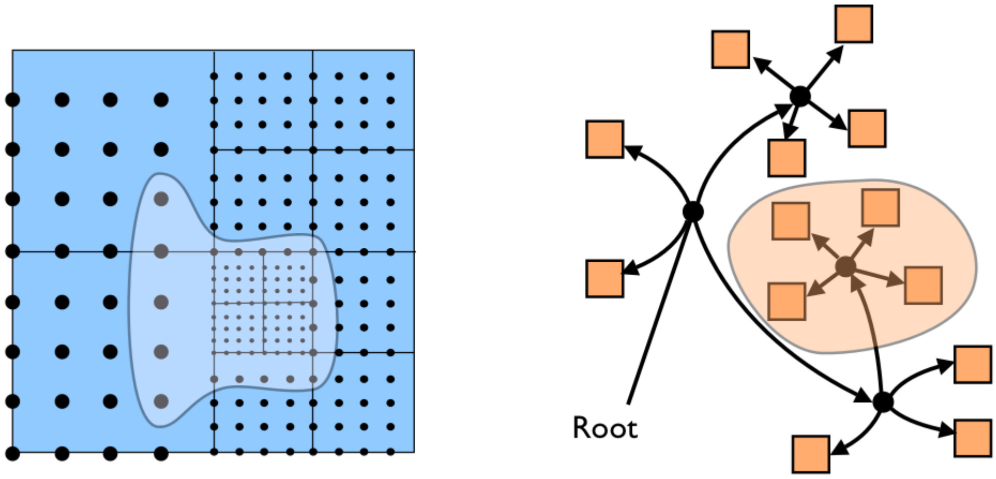


Figure 9.4: Representation of 2D blocks with 4 × 4 scaling coefficients in the physical space of the adapted grid (left) and the associated tree-based data structure (right).

A drawback of storing the lookup table is the large memory required, especially when adjacent resolution jumps larger than one are allowed. For demanding 3D simulations this can increase the memory footprint enough to inhibit high resolution simulations on single or few compute nodes. To provide an alternative for such cases, we implemented the option to compute ghost points on the fly, bypassing the need to store precomputed ghost-point reconstruction patterns. Although this greatly reduces the memory footprint of our solver, it comes at the cost of an extra computational overhead and is therefore beneficial only when simulations are restricted by available memory.

9.3.2 Structured blocks

Wavelet-adapted grids constitute a dyadic tree structure whose leaves are the individual grid points. Such fine-grained trees can achieve large compression rates, but the large number of internal nodes leads to detrimental performance issues. Traversing such a tree involves many strictly sequential operations and memory indirections.

To expose more parallelism and simplify the data structure, scaling coefficients are grouped into fixed-size structured blocks as sketched in figure 9.4 (left), at the expense of a decreased compression rate. The grid is then represented with a tree containing blocks as leaves (figure 9.4, right), rather than individual grid points. In this work blocks contain 32^2 points in 2D and 16^3 points in 3D.

The introduction of blocks brings a number of major performance benefits. Firstly it provides a better usage of the memory bandwidth, since the grid points in a block are represented contiguously in memory and the evaluation of the RHS and FWT rely mostly on unit-stride accesses. Secondly it drastically decreases the memory indirections involved in the tree traversal, since the block-based grid tree contains $\mathcal{O}(10^3)$ less nodes than the tree based on single grid points. Thirdly it enables both explicit and implicit vectorization for the block-based operations, since the scaling coefficients within a block all have the same resolution. Finally, in the following paragraph we present how multithreading can be used for efficient parallel processing on such block-structured adaptive grids.

9.3.3 Thread-level parallelism and work-stealing

The computational work associated to a single block, such as an FWT or RHS evaluation, is assigned exclusively to one thread. This strategy poses two potential limitations on the parallel performance. First, the coarse-grained nature of the blocks limits the amount of parallel work. Second, a work imbalance between the blocks is introduced since the computational cost of ghost reconstruction depends on the local structure of the grid.

Those issues are addressed by considering task-schedulers based on work-stealing (Blumofe, Joerg, et al. 1996; Blumofe and Leiserson 1999) to balance the work on each core. As soon as one parallel worker finishes its work, it steals tasks from the queue of another worker. Work-stealing is used both to address the work imbalance in the ghost reconstruction as well as to address irregular computational patterns of the tree code. The performance of this approach is detailed in Hejazialhosseini, Rossinelli, Bergdorf, et al. (2010), Rossinelli, Hejazialhosseini, Bergdorf, et al. (2011), and Hejazialhosseini, Rossinelli, Conti, et al. (2012).

9.4 Multiresolution remeshed vortex method for FSI

We combine the above multiresolution framework with the remeshed vortex method (rVM) for FSI described in chapter 5. Here we discuss the combination with the rVM-FSI algorithm presented earlier. A major difference between the uniform resolution solver of the previous part, and the current multiresolution solver is the approach to the velocity computation. Whereas before we used FFT-based Poisson solvers, here we have to choose a velocity solver compatible with the multiresolution approach. We present and motivate our choice for a multipole-based solver that is detailed further in chapter 10. Lastly we introduce the software for multiresolution remeshed vortex methods.

9.4.1 Algorithm

The simulation consists of a series of N_{adapt} global time steps wrapped in between a grid refinement and compression analysis to adapt the domain to the spatial scales in the flow. Before each series, we perform the FWT on the vorticity and velocity fields and we refine the adapted grid where new scales are expected to emerge. Afterwards we execute N_{adapt} steps of algorithm 2 in section 5.4 without changing the grid structure. The equations in algorithm 2 are solved at the location of the scaling coefficients, in sequence. In each step the velocity field is reconstructed with a multipole-based solver which we detail below. For the integration of the viscous term, an LTS scheme is used, where scaling coefficients are visited multiple times. For the integration of the convection term, we use a multiresolution particle method. After N_{adapt} time steps we perform another FWT and compress the grid by removing the scaling coefficients corresponding to details deemed negligible.

9.4.2 Velocity solver

The Poisson equations (5.20) and (5.21) are solved using an $\mathcal{O}(N \log N)$ multipole method (Barnes and Hut 1986; L. Greengard and Rokhlin 1987). Multipole methods offer a number of computational advantages that we can exploit in our implementation. First, they can handle arbitrarily spaced source and target points. Second, their computational cost scales separately with the number of source and target points, which is relevant since the vorticity field has a compact support, whereas the streamfunction is global. Third, they allow a natural treatment of free-space boundary conditions. And fourth, the granularity of their computational work is relatively fine as target points can be evaluated independently, making the use of GPU-based accelerators particularly attractive.

Multipole methods enable the flexibility of solving for either the streamfunction components individually, by using the Laplacian kernel, or directly for the entire velocity field, by using the Biot-Savart kernel. The 2D multipole-based solver used in chapter 11 uses the latter approach, and for further details on this solver we refer to Rossinelli (2011) and Rossinelli, Hejazialhosseini, van Rees, et al. (2015).

For the 3D vector Poisson equation instead we solve each component of the streamfunction independently. This approach provides a potential reduction in the algorithmic complexity of the velocity solver, since the computational cost for solving all streamfunction components scales with the number of source points in each direction separately. In particular, let the number of non-zero source points for each vorticity component be N_S^x, N_S^y and N_S^z, respectively, and let the number of target points be N_T. The computational cost for the Biot-Savart kernel is now given as $\mathcal{O}(3N_T \log(\max(N_S^x, N_S^y, N_S^z)))$, where in 3D the base of the logarithm is 8. Solving each component separately scales as $\mathcal{O}(N_T(\log N_S^x + \log N_S^y + \log N_S^z))$. In the most extreme case, where the flow field is effectively 2D and the vorticity field has only one non-zero component with N_S non-zero source points, the speedup is $1 + \log(3)/\log(N_S)$. A potential drawback is the need to take the curl of the streamfunction $\boldsymbol{\Psi}$ to obtain the velocity field, but this can be done accurately and efficiently using finite-difference schemes as discussed above. Another overhead is introduced as the source tree needs to be constructed separately for each component, but it will be shown in the next chapter that this cost is small compared to the total execution time.

Using this approach, we have developed a 3D multipole method to solve for the streamfunction on block-structured multiresolution grids. Its algorithm, implementation and performance are discussed in chapter 10.

9.4.3 Software

Details of the 2D software MRAG-I2D can be found in Rossinelli, Hejazialhosseini, van Rees, et al. (2015). The 3D code developed here is based on a prototype code from Rossinelli (2011). It is written in C++11, except for the multipole-based solver detailed in chapter 10, which contains explicitly vectorized code using SSE, GPU kernels written in CUDA, and distributed-memory code using the Message Passing Interface (MPI). Well-balanced workloads such as strictly local operations on blocks are parallelized with OpenMP (Board 2008). Task-based parallelism based on the work-stealing principle is implemented with the Intel Threading Building Blocks (TBB) library (Contreras and Martonosi 2008; Robison, Voss, and Kukanov 2008).

CHAPTER 10

High performance CPU/GPU multiresolution Poisson solver

In chapter 9 we described how wavelet-based adaptive grids can be used to solve PDEs and coupled to the remeshed Vortex Method. Here we present the multipole-based solver responsible for velocity reconstruction in the 3D multiresolution rVM. The solver is designed for block-structured grids according to multiresolution approach presented in section 9.2. It is based on the established $\mathcal{O}(N \log N)$ multipole method (Barnes and Hut 1986) with some algorithmic changes that improve the performance on CPU/GPU compute architectures. The solver is used for all 3D multiresolution simulations presented in chapter 12.

This chapter is based on van Rees, Rossinelli, et al. (2014).

10.1 Introduction

Multipole methods are used to decrease the computational cost of the N-body problem encountered in many particle-based simulations with applications such as astrophysics and fluid mechanics. Much effort is devoted to optimizing the performance of multipole-based N-body solvers for arbitrarily spaced source and destination particles, both for massively parallel distributed memory architectures (Rahimian et al. 2010) and for GPUs and GPU clusters (Yokota and Barba 2012).

In fluid mechanics, the N-body problem lies at the heart of traditional vortex methods (Leonard 1980). The kinematic relationship between vorticity and the streamfunction leads to an N-body potential problem that needs to be solved at every time step to recover the streamfunction from the vorticity. In the current work we consider remeshed vortex methods (Koumoutsakos 1997), which combine particle-based advection with regular grid-based data representations. The regular grid is commonly exploited for FFT-based elliptic solvers (Chatelain and Koumoutsakos 2010). However, the straightforward use of FFT-based solvers is hindered in a multiresolution setting, where the grid spacing changes according to the local spatial scales in the flow.

For multiresolution remeshed vortex methods, multipole methods offer a number of computational advantages. First, they can handle arbitrarily spaced source and destination points. Second, their computational cost scales separately with the number of source and destination points. This is relevant here since in general the vorticity field has a compact support, whereas the streamfunction needs to be evaluated globally. Third, they allow for a natural treatment of free-space boundary conditions. And fourth, the granularity of the computational work is relatively fine as every destination point can be evaluated independently, rendering the use of GPU-based acceleration techniques particularly attractive.

In this work we present a multipole-based N-body solver to compute the streamfunction from the vorticity field on a multiresolution grid. As opposed to existing algorithms, we designed our solver to exploit the regular structure of the destination points. We outline the algorithm, present details of our optimizations and show performance results of the heterogeneous CPU/GPU implementation on a single Cray XK7 compute node.

10.2 Governing equations

The remeshed vortex method solves the Navier-Stokes equations in the velocity ($\mathbf{u}$)-vorticity ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) formulation. At every time step, the velocity field needs to be computed from the vorticity field. Using the incompressibility of the velocity field ($\nabla \cdot \mathbf{u} = 0$), one can derive a Poisson equation for the solenoidal vector streamfunction $\boldsymbol{\Psi}$

$$\Delta \boldsymbol{\Psi} = -\boldsymbol{\omega}, \tag{10.1}$$

from which the velocity follows as $\mathbf{u} = \nabla \times \Psi$. Throughout this work we consider one component of the above Poisson equation, and employ free-space boundary conditions to simulate an unbounded flow.

The solution to equation (10.1) can be written as a convolution between Green's function for the Laplace equation and the right-hand side

$$\boldsymbol{\Psi}(\mathbf{x}) = -\int G(\mathbf{x}-\mathbf{x}')\boldsymbol{\omega}(\mathbf{x}')\,\mathrm{d}\mathbf{x}', \tag{10.2}$$

where, in three dimensions,

$$G(\mathbf{x}) = -\frac{1}{4\pi|\mathbf{x}|}. \tag{10.3}$$

Here we will briefly recapitulate the multipole method, based on Barnes and Hut (1986) and following the discussion in L. Greengard and Rokhlin (1987). In the discrete case, for N source points located inside a sphere with radius a, and the ith source point having polar coordinates $(r'_i, \theta'_i, \phi'_i)$ and weight ω_i, we have at polar location $\mathbf{x} = (r, \theta, \phi)$

$$\psi(\mathbf{x}) = \sum_{i=0}^{N} \frac{\omega_i}{4\pi\left|\left|\mathbf{x}-\mathbf{x}'_i\right|\right|} = \frac{1}{4\pi r}\sum_{i=0}^{N} \frac{\omega_i}{\sqrt{1+(r'_i/r)^2 - 2(r'_i/r)\cos\gamma_i}}, \tag{10.4}$$

where γ_i is the angle between the vectors $\mathbf{x}$ and $\mathbf{x}'_i$. If for each source point we have $(r'_i/r) < 1$, we can substitute the definition of Legendre polynomials $P_n(x)$ as Taylor series, and use the addition theory for spherical harmonics $Y_n^m(\theta, \phi)$ to find

$$\psi(\mathbf{x}) = \frac{1}{4\pi}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}\left\{\sum_{i=0}^{N}\omega_i r'^n_i \overline{Y_n^m}(\theta'_i, \phi'_i)\right\}\frac{Y_n^m(\theta,\phi)}{r^{n+1}} \equiv \frac{1}{4\pi}\sum_{n=0}^{\infty}\sum_{m=-n}^{n} C_n^m \frac{Y_n^m(\theta,\phi)}{r^{n+1}}. \tag{10.5}$$

In this expression, the multipole coefficients C_n^m can be precomputed for each collection of sources, and used in the evaluation of equation (10.5) for every destination point.

If the infinite sum over n is truncated to a finite number of terms $p+1$, the following error norm holds

$$\left|\psi(\mathbf{x}) - \frac{1}{4\pi}\sum_{n=0}^{p}\sum_{m=-n}^{n} C_n^m \frac{Y_n^m(\theta,\phi)}{r^{n+1}}\right| \leq \sum_{i=1}^{N}|\omega_i|(r-a)^{-1}\left(\frac{a}{r}\right)^{p+1}. \tag{10.6}$$

Evaluation of the direct interactions requires a discretization of equation (10.2), with the Green's function given in equation (10.3). The singular nature of this integral reduces the accuracy of the scheme close to the singularity. Furthermore, it requires special treatment of the singularity itself, i.e. the case $\mathbf{x} = \mathbf{x}'$, which introduces an instruction irregularity in the code. For these reasons we choose to replace the original Green's function with a smoother function, as derived in Winckelmans and Leonard (1993)

$$G^{\epsilon}(r) = -\frac{1}{4\pi}\frac{r^2 + \frac{3}{2}\epsilon^2}{(r^2+\epsilon^2)^{\frac{3}{2}}}, \tag{10.7}$$

where ϵ is a smoothing parameter. In our simulations, we set $\epsilon = h$, where h is the smallest grid spacing in the domain. This formulation removes the singularity and associated instruction irregularity, and improves the accuracy of the results.

10.3 Computational setting

The multipole-based solver described herein is part of a 3D incompressible flow solver. The solver is built on MRAG, a wavelet-based, multiresolution block-structured framework to solve partial differential equations (Rossinelli, Hejazialhosseini, Spampinato, et al. 2011), and is based on the remeshed vortex method (Koumoutsakos and Leonard 1995). In the remeshed vortex method each time step requires three executions of the multipole algorithm, once for each of the three components of the streamfunction. Applications of the solver include the simulation and optimization of self-propelled swimmers (Gazzola, van Rees, and Koumoutsakos 2012; van Rees, Gazzola, and Koumoutsakos 2013).

In MRAG, the computational grid consists of blocks with a fixed number of gridpoints in each direction (here 16). The computational blocks are non-overlapping leaves of a hierarchical tree structure that represents the computational grid, so that each block can exist at a different spatial resolution. Blocks can be refined (split into new blocks) or compressed (merged with neighboring blocks) based on the magnitude of the detail coefficients in the forward wavelet transform of relevant quantities with respect to user-specified thresholds. Operations across all blocks are parallelized on a multicore node with task-based parallelism, based on the work-stealing principle as implemented in the Intel Threading Building Blocks (TBB) library (Robison, Voss, and Kukanov 2008). More details on MRAG and its parallel implementation on multicore CPUs are given in Rossinelli, Hejazialhosseini, Bergdorf, et al. (2011) and Rossinelli, Hejazialhosseini, Spampinato, et al. (2011).

10.4 Algorithm

The algorithm used here is based on a Barnes-Hut tree code (Barnes and Hut 1986) and adapted for the current computational setting.

We filter all grid points to extract those with non-zero vorticity values, creating an array of source points spanning only the support of the vorticity field. These source points are hierarchically decomposed into an oct-tree structure, guided by a parameter controlling the maximum number of source points per leaf ($s_{\max}$, here set to 2048). For each leaf the multipole expansion coefficients C_n^m are calculated according to equation (10.5). At parent nodes the expansions of the children are combined and translated to the parent centers (L. Greengard and Rokhlin 1987). All expansions are stored in an array.

We group a set of destination points into a *brick* (here 8 grid points in each direction) to increase the amount of work per task. All points in a destination brick are subjected to the

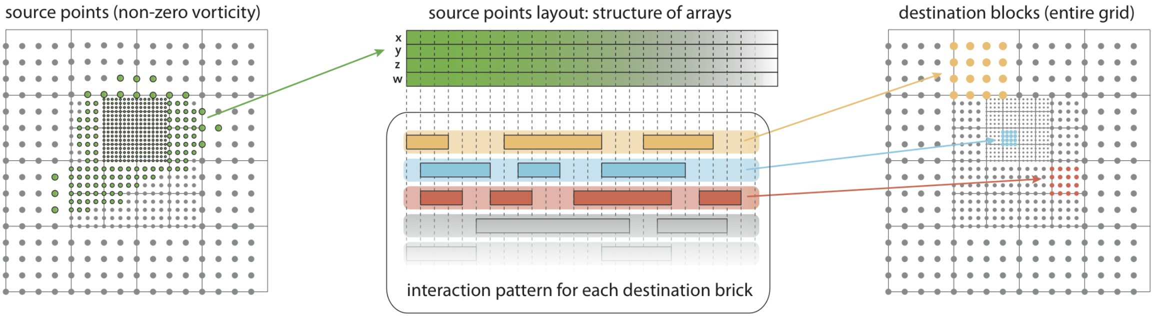

source points (non-zero vorticity)
source points layout: structure of arrays
x
y
z
w
interaction pattern for each destination brick
destination blocks (entire grid)

Figure 10.1: Logical plan for the direct interactions. The set of grid points with non-zero vorticity (left) is converted to a series of Structure of Arrays (SoA). For each structured, fixed-size destination brick the plan defines a list of pairs representing the start and end indices (inclusive) of the source points to interact with. For simplicity, destination bricks in this sketch are represented as 2D blocks of 4 × 4 grid points.

same logical interaction pattern. For each destination brick, therefore, we create two logical plans containing the lists of all direct and all indirect interactions, respectively. Separating the creation of the plan from the actual interaction evaluations enables offloading of all direct interactions to an accelerator such as the GPU.

The logical plans are based on a user-specified opening parameter θ. Specifically, for each destination brick, we traverse the source tree downwards. Starting at the root node, we compare the node radius a (the radius of the smallest sphere containing its sources) with the smallest distance r between the node center and the points inside the destination brick. If $a/r < \theta$, we will perform a multipole evaluation according to equation (10.5) between the tree node and all points in the destination brick. If $a/r \geq \theta$, we traverse one level down the tree and repeat the process for all the node's children. If we reach a leaf, we perform direct evaluations according to equation (10.4) between all points in the source node and all points in the destination brick. In this way, if $\theta = 0$ we perform the $\mathcal{O}(N^2)$ problem whereas for $0 < \theta < 1$ we have a converging multipole approximation with truncation error bound by equation (10.6), as a function of the order of the multipole expansion p and the opening parameter θ.

The logical plan for the direct evaluations of a destination brick consists of a vector of pairs, where each pair encodes the start and the end index of a range in the source point arrays (see figure 10.1). For the indirect evaluations, the logical plan consists of a vector of tuples, where each tuple contains the 3D center of the expansions and the index of the expansion in the expansions array.

Finally, we evaluate the logical plans for each destination brick and sum the results to obtain the streamfunction field.

10.5 Implementation

The algorithm is implemented in C++11 and is parallelized using OpenMP tasks and the Threading Building Blocks (TBB) library. The order of the multipole expansion p, is a compile-time constant that is fixed at $p = 6$ in this work. In this section we will briefly discuss the implementation of the key parts of the algorithm.

Source points The source points are converted from an Array-of-Structures (AoS) to a Structure-of-Arrays (SoA) format to enable vectorization over the source particles in the later stages of the solver. To increase spatial locality, we sort the particles in Morton order with the OpenMP-based GNU libstdc++ parallel sort.

Tree construction and logical plan The oct-tree containing all source particles is recursively constructed using OpenMP tasks. We first perform a top-down pass to identify and create the nodes on each level, and then perform a bottom-up pass to compute the multipole coefficients C_n^m according to equation (10.5). After the tree construction, the logical plan for the evaluation phase is created using TBB parallel tasks across the destination bricks.

Legendre polynomials P_n^m The associated Legendre polynomials $P_n^m(\cos\theta)$ are evaluated for $0 \le n \le p$, $0 \le m \le n$ through the recursive equations

$$\begin{aligned}
&P_0^0 = 1, \qquad P_1^0 = \cos(\theta), \qquad P_1^1 = \sin(\theta) && \\
&P_n^0 = \frac{(2n-1)}{n}\cos(\theta)P_{n-1}^0 - \frac{(n-1)}{n}P_{n-2}^0 && 2 \le n \le p, \\
&P_n^m = (2n-1)\sin(\theta)P_{n-1}^{m-1} + P_{n-2}^m && 2 \le n \le p, \quad 1 \le m \le n-2, \\
&P_n^m = (2n-1)\sin(\theta)P_{n-1}^{m-1} && 2 \le n \le p, \quad n-1 \le m \le n.
\end{aligned}$$

We store the polynomials in a linear data structure of size $(p+1)(p+2)/2$. All recursions are templatized over n, and for each n in turn over m, so that all instruction irregularity is resolved at compile-time and array access indices can be precomputed. Furthermore, the associated Legendre polynomials are vectorized with SSE instructions so that the values of P_n^m can be computed for four different arguments in one set of operations.

Spherical harmonics Y_n^m The spherical harmonics $Y_n^m(\theta,\phi)$ are defined as

$$Y_n^m(\theta,\phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}}P_n^{|m|}(\cos\theta)\left(\cos(m\phi) + i\sin(m\phi)\right),$$

for $0 \leq n \leq p$ and $-m \leq n \leq m$. To reduce the number of costly trigonometric evaluations, we compute the sine and cosine factors recursively using the identity

$$\begin{aligned}\sin(m\phi) &= \sin(\phi)\cos((m-1)\phi) + \cos(\phi)\sin((m-1)\phi),\\ \cos(m\phi) &= \cos(\phi)\cos((m-1)\phi) - \sin(\phi)\sin((m-1)\phi).\end{aligned}$$

Finally we note that the prefactor in the definition of the spherical harmonics, for $m > 0$, can be rewritten as

$$\frac{(n-m)!}{(n+m)!} = \frac{1}{(n+m)(n+m-1)(n+m-2)\ldots(n-m+1)}.$$

This formulation reduces the number of floating point evaluations and is precomputed for all $n \leq p$ and $m \leq n$ at compile-time.

The spherical harmonics are stored into two arrays, each of size $p(p+1)/2$, for the real and complex part, respectively. Again we vectorize the computation using SSE so that we can compute the set of all harmonics for four different arguments (θ, ϕ) in parallel.

Multipole coefficients C_n^m When creating the multipole coefficients C_n^m from a set of source particles, we use SSE instructions to process four source points in parallel. Again template recursion allows for unrolled instructions over the n and m indices. The values C_n^m are stored in three arrays: two of size $p(p+1)/2$ for the real and imaginary part of C_n^m, $m > 0$, and one more of size $(p+1)$ for the $m = 0$ values. Based on the multipole coefficients for a leaf node of our source tree, we can use translation equations to find the coefficients at parent nodes (L. Greengard and Rokhlin 1987). In this translation computation we use a precomputed look-up table for those prefactors known at compile-time.

CPU-only evaluation When evaluating both indirect and direct interactions on the CPU, we rely on three levels of TBB-based parallel operators to exploit the multicore architecture. The first level of parallelism covers all the bricks, the second covers the direct and the indirect interactions, and the third covers the interactions themselves. At the finest level of parallelism we therefore evaluate a (sub)set of interactions (direct or indirect) for all the destination points inside a brick.

The direct evaluation kernel on the CPU is vectorized with AVX instructions. Since our bricks consist of 8 grid points per dimension, this allows us to evaluate all the grid points along the x-direction in parallel. Based on equation (10.7), we have 15 floating-point operations and 1 reciprocal square root per particle-particle interaction. We implement the reciprocal square root with the AVX native approximation and a further Newton-Raphson

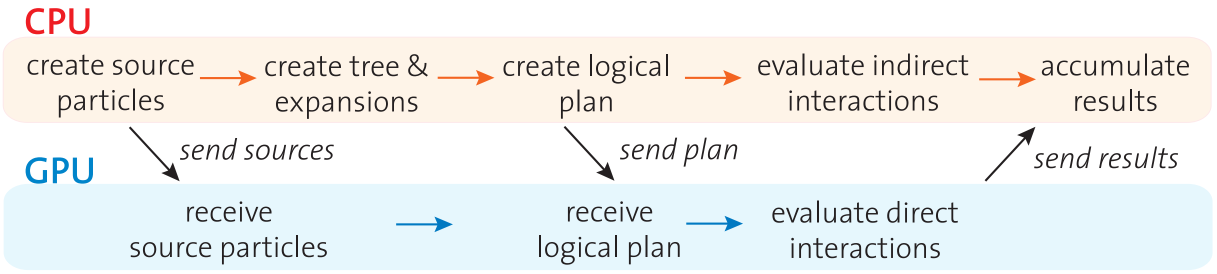


Figure 10.2: Hybrid CPU/GPU workflow, showing the steps on the CPU (red) and GPU (blue) for one streamfunction evaluation.

iteration, leading to a total cost of 21 floating point operations. Out of these 21 operations, 4 are Fused Multiply-Add (FMA) operations.

The indirect evaluation kernel on the CPU is currently vectorized with SSE instructions over the destination points along the x-direction. These instructions are unrolled at compile-time with templates. Here we count the number of instructions directly from the assembly code, which gives 46 floating point operations per particle-multipole interaction (for $p = 6$), out of which 11 are FMA.

Hybrid CPU/GPU evaluation In case we use the GPU to evaluate the direct interactions, we send the source data to the GPU right after creating the SoA source arrays, so that the transfer overlaps with the next steps on the CPU: computing the multipole coefficients and creating the logical evaluation plans. After the plans and the tree are created, we send the logical plan for the direct evaluations to the GPU and start the evaluation asynchronously, so that it is executed in parallel with the evaluation of the indirect interactions on the CPU. Finally, we transfer back the results from the GPU (figure 10.2).

The GPU implementation maps all direct interactions for a single destination brick on one streaming multiprocessor (SM), so that each CUDA block is a 3D array with 8 threads in each direction. If the number of bricks exceeds the maximum CUDA grid size, we perform multiple passes. We use the shared memory on an SM to load up to 64 source point locations and weights in parallel, and subsequently each CUDA thread in the SM evaluates those source points for its own destination point. The interaction kernel performs 15 FLOPs (including 3 FMA) and 1 reciprocal square root with a cost equivalent to 6 FLOPs (Corporation 2012).

The amount of direct interactions to be computed varies between the bricks, which could lead to load-imbalance. However, we found that the hardware scheduler on the GPU is effective in distributing the work among the available SMs and we did not observe significant load-imbalance problems.

Distributed-memory parallelism Although the current chapter deals entirely with single-node computations, a simple master/slave distributed-memory extension was developed. The extension is based on MPI and was used for the results of chapter 12 in order to reduce the time-to-solution.

The distributed-memory layer was designed as in Rossinelli (2011), wrapping around the software as discussed in this chapter. In this master/slave model, the master rank computes the tree, multipole expansions, interaction plan and brick distribution as before. It broadcasts the entire source arrays and all multipole expansions to all slaves. The master distributes the target bricks among all ranks so that each will compute approximately the same number of indirect interactions. This approach equally balances the workload since the indirect interactions dominate the time to solution, as discussed in the next section. The master rank sends the appropriate interaction lists and brick information structures to each of the slaves. All ranks then evaluate the direct and indirect interactions according to their assigned set of bricks, using the high-performance software presented in the this chapter. Finally, the results are gathered by the master rank and the computational grid is updated.

Simulations in chapter 12 are typically performed on four compute nodes. Since the granularity of the distributed-memory parallelism layer is in terms of bricks, the load-balancing can be performed very accurately. As a result, the difference in compute times between the ranks is less than 0.5%. However the overheads due to sending and receiving data with this algorithm can take up to 15% of the total evaluation time. For this reason future work should investigate reducing the granularity on this layer from bricks to blocks, to increase the scalability of this distributed-memory solution.

10.6 Results

We run our performance tests on a single 16-core AMD Interlagos 6272 compute node, with a NVIDIA Tesla K20X GPU, as available on the Cray XK7 system “Tödi” at the Swiss Supercomputing Center (CSCS). The peak performance of all 8 FPUs of this compute node is 268.8 GFLOP/s, and the listed peak performance of the K20X is 3.95 TFLOP/s. Both the CPU and GPU support FMA instructions, so the upper performance bounds of our kernels are adjusted according to their FMA/non-FMA ratios. The code is compiled with version 4.7 of the GNU C++ compiler and version 4.1.0 of TBB.

We consider two test cases: the first is an artificial constructed test problem while the second considers a relevant flow problem.

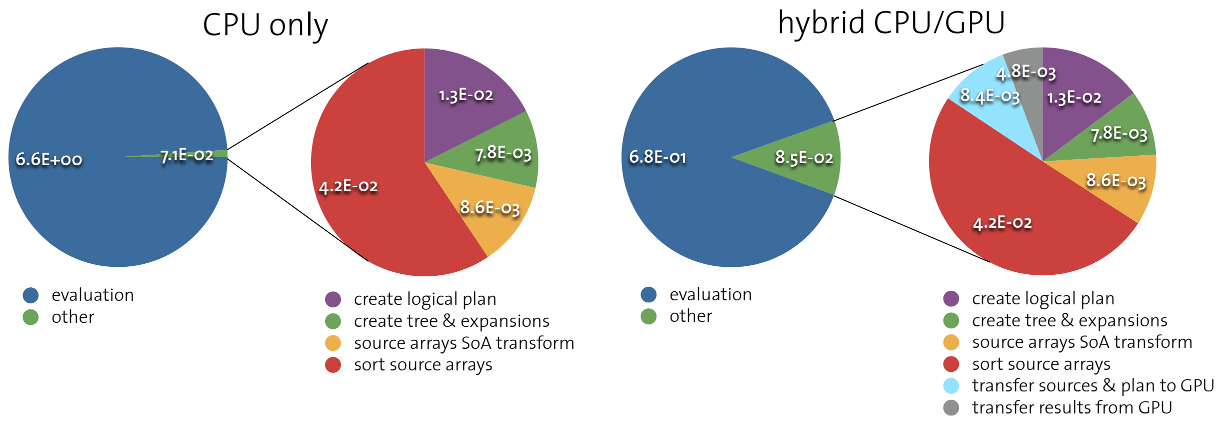


Figure 10.3: Time distribution (in seconds) of the presented algorithm with $\theta = 0.5$, CPU-only (left) and hybrid CPU/GPU (right).

10.6.1 Test problem

We consider a test problem introduced in Hejlesen, Rasmussen, et al. (2013), consisting of a vortex ring with radius R, its streamfunction defined as

$$\boldsymbol{\Psi}(r,z) = f\left(\frac{\sqrt{(r-R)^2+z^2}}{R}\right)\mathbf{e}_\theta, \quad \text{where} \quad f(t) = \begin{cases} c_1 \exp\left(-\dfrac{c_2}{1-t^2}\right) & \text{if } |t| < 1, \\ 0.0 & \text{else.} \end{cases}$$

The vorticity is derived from the streamfunction as $\boldsymbol{\omega} = \nabla \times (\nabla \times \boldsymbol{\Psi})$. We place a ring at $(0.25, 0.25, 0.25)$ and at $(0.75, 0.75, 0.75)$ in a unit cube domain with origin $(0,0,0)$. For each ring, we set $R = 0.125$, $c_1 = 2^{27}$ and $c_2 = 20$. We rotate one of the rings by $\pi/2$ so that the axes of the rings are not aligned. The effective resolution of the grid is 256^3, with compression and refinement thresholds of 10^{-4} and 10^{-2}, respectively, resulting in 3.6×10^6 destination points. We consider the y-component of the streamfunction, for which we have 8.7×10^5 source particles.

In figure 10.3 we show the time distribution of the CPU implementation (left) and CPU/GPU implementation (right) for the case of $\theta = 0.5$. The evaluation phase, which covers both the direct and indirect interactions, takes most of the time for both cases. The remaining time is spent mostly in sorting the source particles, after that it is about equally spent in the remaining parts of the algorithm.

Since the evaluation phase takes most of the time, we focus our analysis there. Strong parallel scaling of the evaluation step on the CPU reaches 96.7% on 8 cores, in which case all FPUs are active. Doubling the number of cores to 16 brings only very small improvements for the evaluation phase, although other parts of the solver, notably the creation of the logical plan, benefit from these additional cores.

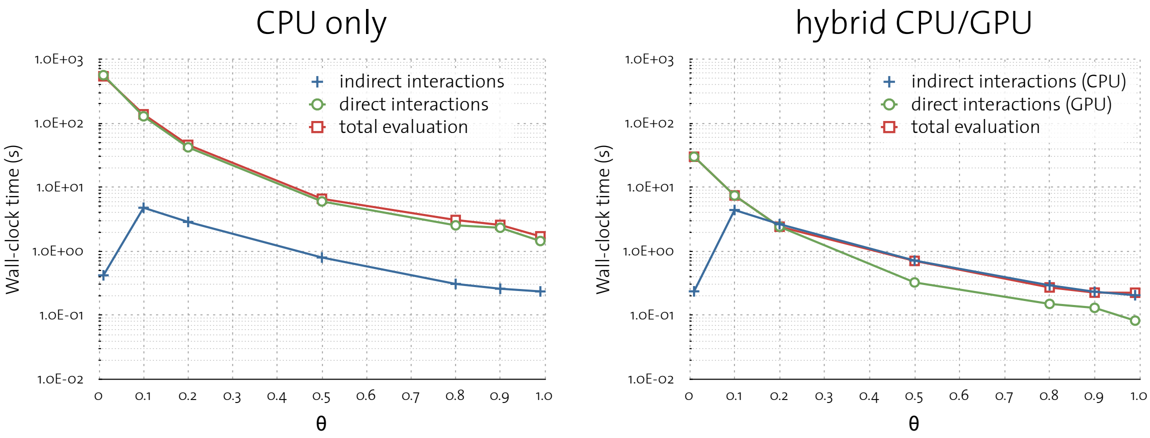


Figure 10.4: Time spend in computing direct and indirect interactions as a function of θ, CPU-only (left) and hybrid CPU/GPU (right).

Details about the evaluation phase are shown in figure 10.4. For the parallel parts of the code, we accumulate the timings measured for each brick and divide this number by the total number of threads (16), to obtain a measure of the wall-clock time spent in each phase of the evaluation. We plot the timings against different values of θ. The time spent per interaction evaluation remains approximately constant for both the direct (CPU and GPU) and the indirect evaluations.

For the CPU-only execution, across the entire range of θ values, we spend at least one order of magnitude more time for the direct interactions than for the indirect interactions. Offloading the direct interactions to the GPU achieves a speedup of about 17 in their evaluation time, so that for $\theta \gtrsim 0.2$ their computation is effectively hidden and the solution time is dictated by the evaluation of the indirect interactions on the CPU. Table 10.1 shows that indirect and direct interactions on the CPU achieve 81.4% and 74.1% of the upper performance bound, respectively, while the direct interactions on the GPU achieve 96.5% of the upper performance bound.

Table 10.1: Performance of CPU and GPU components for the evaluation of the test problem.

	CPU (GFLOP/s)		GPU (TFLOP/s)	
	measured	upper bound	measured	upper bound
direct interactions	118.6	160.0	2.18	2.26
indirect interactions	135.6	166.5	n/a	

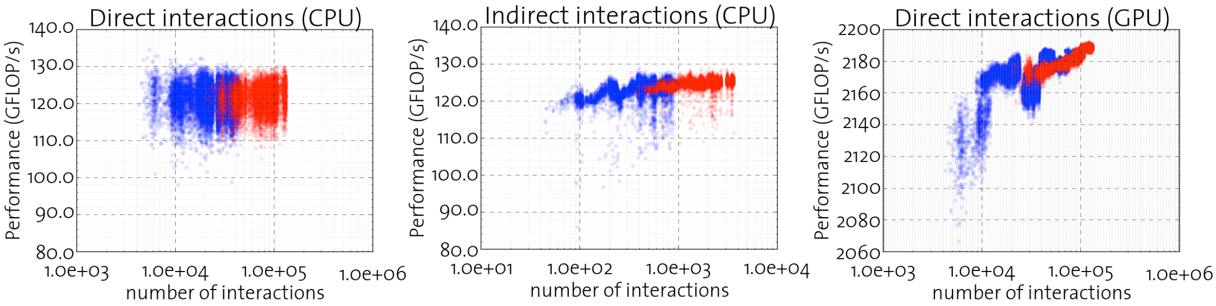


Figure 10.5: Performance measurements from simulations of the flow past a sphere, with effective resolution of 512^3 (blue) and 1024^3 (red).

10.6.2 Flow past a sphere

We report performance results of our software for the flow past a sphere at Reynolds number 550, with effective resolutions of 512^3 and 1024^3, and $\theta = 0.5$. The performance of each streamfunction component evaluation within the first 2500 time steps (up to non-dimensional time $T = 1.5$) is plotted in figure 10.5. The x-axis represents the number of interactions, which varies as the vorticity support changes, and as the grid adapts according to the flow scales.

The performance for both direct and indirect interactions on the CPU shows little variation with the number of interactions, whereas on the GPU performance increases slightly with increasing number of interactions. Presumably the reason is a more effective load-balancing, as there are more degrees of parallelism when the number of bricks increases. The measured performance is very close to the numbers reported for the test problem above, showing that also during a production simulation we can sustain high fractions of the upper performance bound.

We note that, as the number of source and destination points increase, the difference between CPU and GPU computing times increases. This could be a motivation to increase the ratio between direct and indirect interactions, for instance by decreasing θ.

10.7 Conclusions

We presented a hybrid CPU/GPU multipole-based N-body solver for multiresolution grids. We have provided a detailed description of the equations, the algorithm and the optimizations performed to maximize performance. The software achieves approximately 90% of the upper performance bound for the most time-consuming phase of the algorithm. Offloading the direct interactions to the GPU allows us to harness its approximately 15 times larger peak performance, while freeing the CPU to perform the more complicated indirect interactions in parallel. In practice, we observe similar compute times for the CPU and the GPU, although a more fine-tuned approximation of the parameters θ and p could maximize the overlap while still meeting a user-specified accuracy.

The algorithm will be used for flow optimizations of bluff-body flows and self-propelled swimmers. Future improvements of the Poisson solver will focus on using AVX instructions for the indirect interactions, to anticipate its use on the Cray XC30.

Acknowledgments We thank Dr. Peter Messmer (NVIDIA) for several helpful discussions on improving the performance of the direct interaction evaluations on the GPU.

CHAPTER 11

Self-propulsion of a counter-rotating cylinder pair in a viscous fluid

In this chapter we apply the two-dimensional multiresolution flow solver (Rossinelli, Hejazialhosseini, van Rees, et al. 2015) to the study of a self-propelled rotating cylinder pair. We investigate the behavior and analyse the different flow regimes as a function of non-dimensional cylinder width and rotational Reynolds number.

11.1 Introduction

Steady locomotion of aquatic swimmers, involves a body translating with respect to the surrounding fluid with a net zero hydrodynamic force. Natural swimmers at intermediate and high Reynolds numbers achieve this by alternating periods of thrust and drag through periodic body deformations such as undulations in fish or contractions in jellyfish (Childress 1981; M. S. Triantafyllou, G. S. Triantafyllou, and Yue 2000; Wu 2011). The physical mechanisms underlying these forms of propulsion are widely studied and mimicked for relevant engineering applications (M. S. Triantafyllou and G. S. Triantafyllou 1995; Lauder et al. 2007; Whittlesey, Liska, and Dabiri 2010). Alternative forms of aquatic locomotion have been presented for a flapping body moving laterally due to vortical instabilities (Alben and Shelley 2005), and for a passive body in a flow due to fluid mediated interactions of a nearby actively controlled object (Gazzola, Mimeau, et al. 2012).

Here we consider a simplified concept for locomotion formed by two steadily counter-rotating cylinders that are linked together by their centers of mass. This simple system is characterized by two parameters, the non-dimensional width and the rotational Reynolds number of the cylinders. The propulsion mechanism relies purely on viscous effects, since in potential flow d'Alembert's paradox states the force on the body is zero at all times.

Similar studies have been previously reported for two counter-rotating cylinders (Sungnul and Moshkin 2009; Suwannasri 2013) and a 3D axisymmetric torus (Moshkin and Suwannasri 2012). In these works the authors numerically solved for the steady-state flow, and found that two directions of motions can be obtained for specific surface rotation rates and object geometries. Furthermore, the counter-rotating cylinder pair was studied in the absence of self-propulsion, by fixing the cylinders' location in space and applying a uniform inflow velocity (Chan et al. 2011). For certain combinations of inflow and rotational velocities, the vortical wake was suppressed resulting in zero or negative drag values. Both studies confirm the existence of two flow regimes. When the inner surfaces of the cylinders spin against the inflow direction, an elliptical region of circulating fluid around the entire pair can be established, whereas in the opposite rotational direction closed streamlines around each of the cylinders are found. However, the unsteady locomotory behavior of the cylinder pair, as well as a systematic investigation of the cylinders' behavior in the space of governing parameters, have not been reported yet.

Here we study the self-propelling cylinder pairs through unsteady simulations using a 2D multiresolution remeshed vortex method. This enables the study of transient behaviour and the emergence of instabilities in the system. Furthermore we explore a wide phase space for the rotation and width of the pair. Using streamlines and passive tracer particles we discuss the governing flow physics and report on the high Reynolds number instabilities.

In section 11.2 we describe the computational setup, followed by an overview of the results (section 11.3), a discussion of our findings from a hydrodynamic point of view (section 11.4) and the conclusions in section 11.5.

11.2 Setup and methods

In this section we first explain the problem setup and then introduce the computational method used for the flow simulations.

11.2.1 Problem setup

We consider two identical counter-rotating cylinders that are virtually connected by their respective centers of mass so as to move as a single rigid body, as sketched in figure 11.1. The geometry is defined by the distance between the centers of mass of the cylinders (W) and the diameter of the cylinders (D). The cylinders rotate with equal but opposite angular velocity Ω, with a corresponding circulation of strength $|\Gamma| = \pi|\Omega|D^2/2$.

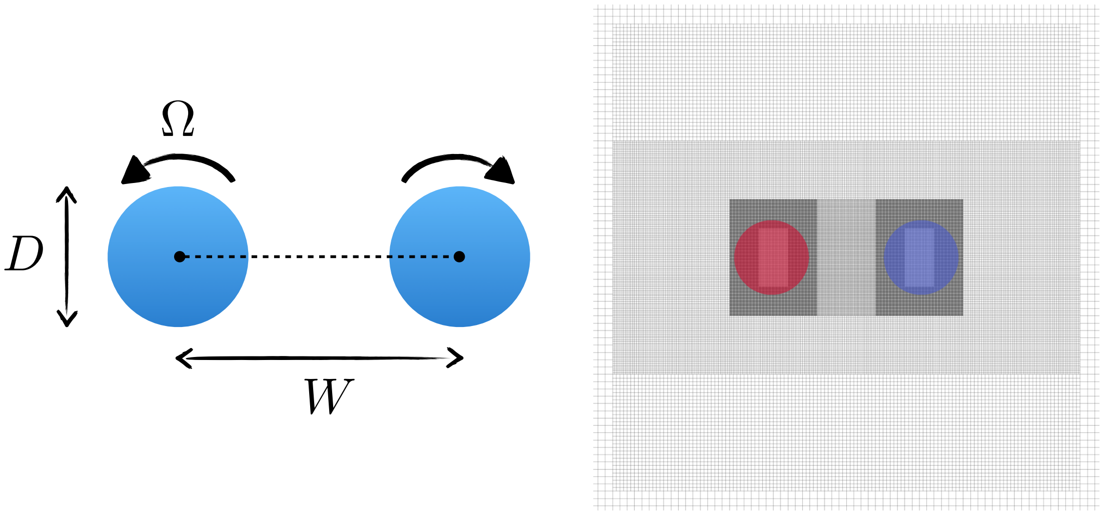


Figure 11.1: Sketch of the problem setup (left) and a close-up of the initial wavelet-adapted grid with the two cylinders colored in red and blue respectively (right). The problem is governed by two non-dimensional numbers, the Reynolds number $Re_\Gamma = \Gamma/\nu = \pi\Omega D^2/(2\nu)$ and the non-dimensional width $W^* = W/D$.

This body is immersed in a fluid with viscosity ν and density ρ_f, which is equal to the body density ρ_s so that gravitational effects can be ignored. Each setup can be uniquely characterized by the non-dimensional width $W^* = W/D$ and the Reynolds number $Re_\Gamma = |\Gamma|/\nu$, where ν is the kinematic viscosity of the fluid. The alternative Reynolds number definition $Re_\Omega = |\Omega|D^2/\nu = \pi Re_\Gamma/2$ can be easily obtained from Re_Γ. Throughout this work we set a positive (counter-clockwise) angular velocity for the left cylinder. Time t is non-dimensionalized using the period of revolution of the cylinders, $T = t|\Omega|/(2\pi) = t|\Gamma|/(\pi^2 D^2)$, and the velocity u is non-dimensionalized using the surface velocity so that $U = 2u/(|\Omega|D) = u\pi D/|\Gamma|$.

11.2.2 Computational Method

All simulations are performed with a 2D multiresolution remeshed vortex method using penalization and projection methods to account for the fluid-solid interaction (Gazzola, Chatelain, et al. 2011). The method solves the vorticity equation on a wavelet-adapted computational grid, using a multipole-based solver to reconstruct the velocity from the vorticity field (Rossinelli, Hejazialhosseini, van Rees, et al. 2015). Particles are used for the advection part of the equation, diffusion is handled with a fourth-order finite difference scheme and an explicit Runge-Kutta 2 time stepping scheme. The solver uses local time stepping whenever the diffusion time step restriction dominates. The method and code have been extensively used and validated for similar problems (Gazzola, Chatelain, et al. 2011; Gazzola, Mimeau, et al. 2012; Gazzola, van Rees, and Koumoutsakos 2012; van Rees, Gazzola, and Koumoutsakos 2013; Rossinelli, Hejazialhosseini, van Rees, et al. 2015). Throughout this work we set the effective resolution of our unit square domain to $ER = 32\,768^2$, and the cylinder radius equal to $R = 0.0025$, so that there are over 80 grid points across the radius. The refinement and compression thresholds are $r_{\text{tol}} = 10^{-3}$, $c_{\text{tol}} = 10^{-4}$, and the cylinder is represented on the computational grid through a discrete Heaviside function (Towers 2009). The time step criterion is $\text{LCFL} = 0.05$ for all results presented here, and we ramp up the circulation strength of the cylinders with a quarter period of a sine function during their first rotation.

11.3 Results

First we show the result of varying of Re_Γ while keeping $W^* = 2$, then we vary W^* while fixing $Re_\Gamma = 100$. Subsequently we show the characteristic behavior of the cylinders as a function of both Re_Γ and W^*.

11.3.1 Varying Re_Γ with $W^* = 2$

The vertical component of the translational body velocity for different Reynolds numbers at a fixed non-dimensional width of $W^* = 2$ is shown in figure 11.2. For Reynolds numbers up to $Re_\Gamma = 85$ the cylinders move upwards, reaching ever higher steady-state velocities as the Reynolds number increases. At $Re_\Gamma = 90$ the cylinders accelerate upwards for the first five revolutions but then the acceleration gradually changes direction and remains negative, so that the cylinders eventually move in downward direction and reach a negative steady-state velocity. For higher Reynolds numbers the cylinders are consistently moving downwards, reaching higher steady-state speeds and larger initial acceleration magnitudes as the Reynolds number increases. The right plot in figure 11.2 shows the vertical velocity of the cylinders for Reynolds numbers above $Re_\Gamma = 100$. We can observe that the downwards velocity magnitude increases until $Re_\Gamma = 500$ but oscillations in the velocity curve appear. At $Re_\Gamma = 625$ and $Re_\Gamma = 750$ the cylinders oscillate strongly upwards and downwards without reaching steady-state within the first 125 revolutions.

Representative vorticity fields corresponding to the upwards ($Re_\Gamma = 50$) and downwards ($Re_\Gamma = 150$) regime are shown in figure 11.3. At steady-state the vorticity field at $Re_\Gamma = 50$ is mostly contained around the cylinders, and the strongest vorticity values are those corresponding to the cylinders' rotations. In contrast, at $Re_\Gamma = 150$ a wake is clearly visible corresponding to a strong flow in between the cylinders, and the highest vorticity magnitudes are just outside of the cylinder surfaces on the inside of the pair.

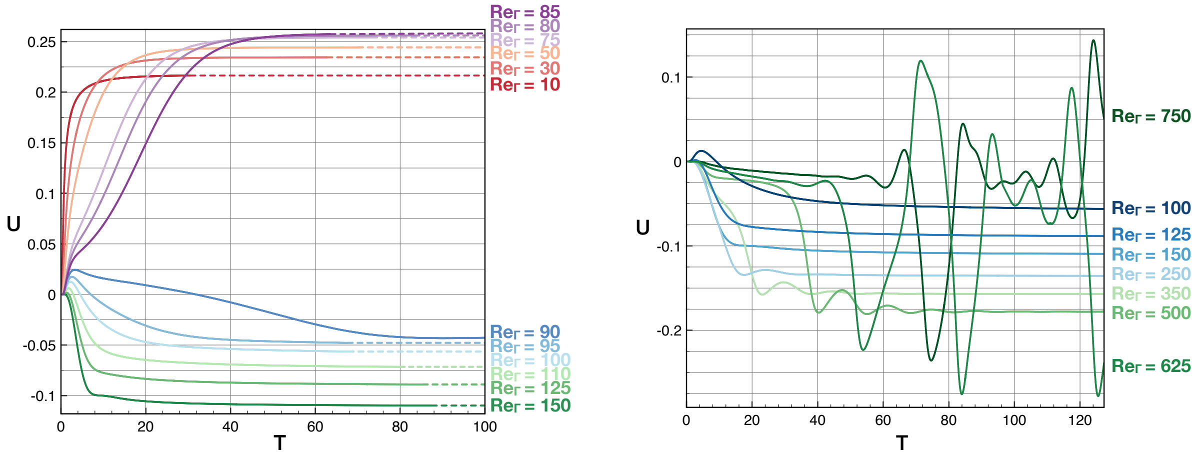


Figure 11.2: Velocity of the counter-rotating cylinder pair at $W^* = 2$ at a series of Reynolds numbers up to $Re_\Gamma = 150$ (left) and up to $Re_\Gamma = 750$ (right). For the left plot, simulations are stopped whenever the velocity does not change more than 0.03% in the last time unit, and extended with a dashed line to indicate the steady-state value.

11.3.2 Varying W^* with $Re_\Gamma = 100$

In figure 11.4 we present the vertical velocity curves for a fixed Reynolds number of $Re_\Gamma = 100$ while varying the non-dimensional width W^*. In this case the cylinders move upwards for $W^* \leq 1.75$ and downwards for $W^* \geq 2$. Increasing the value of W^* even further results in unstable behavior as the cylinders alternate between upwards and downwards motions.

11.3.3 Phase-space results

To quantify the transition between upwards and downwards motion, as well as between stable and unstable motion, we performed a series of computations varying both Re_Γ and W^*. Each computation is run until steady-state and the trend is noted. The points in the Re_Γ–W^* space are chosen manually, both to accurately capture the change in motion direction and to cover the parameter space. A flow case is determined unstable if oscillations in the velocity evolution are observed that do not dampen out during the first $T = 100$ cylinder revolutions.

The results, presented in figure 11.5, show that we the transition between upwards and downwards motion occurs at lower Re_Γ values when the non-dimensional width is increased. Similarly, for small values of W^* the transition happens at very large Reynolds numbers. The extreme cases, towards very large or very large widths, could not be simulated due to

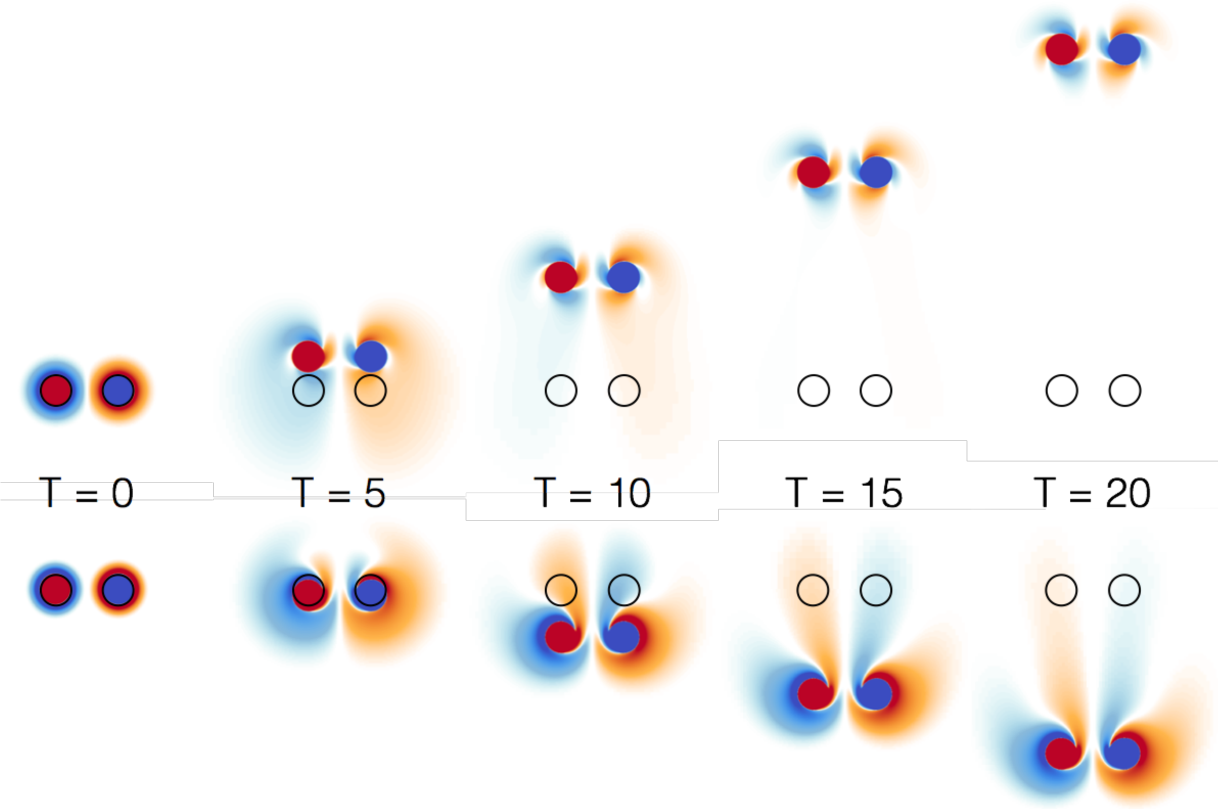


Figure 11.3: Vorticity field of the counter-rotating cylinder pair at $W^* = 2$ at $Re_\Gamma = 50$ (top) and $Re_\Gamma = 150$ (bottom, separate simulation) at different times. The initial location of the cylinder pair is drawn with solid black circles. Animations are provided in the supplementary material.

computational restrictions, however for any width within $1.25 \leq W^* \leq 10$ we found the upward/downward transition occurring at a finite Reynolds number.

The boundary between upwards and downwards motion is characterized by an exponential relationship. The transition points, here defined for each W^* as the average between the largest Re_Γ resulting in upwards motion, and the smallest Re_Γ resulting in downwards motion, are used for a least squares fit with the function

$$Re_\Gamma = a \exp\left[\left(\frac{b}{(W^* - 1)}\right)^c\right].$$

This results in the parameters $a = 6.7$, $b = 17.5$ and $c = 0.33$.

The instability is also characterized by an exponential function of W^*, in this case we find $a = 13$, $b = 56$ and $c = 0.33$. Here the fit deviates further from the data point, as the onset of an instability is harder to identify. Nevertheless again within the range of

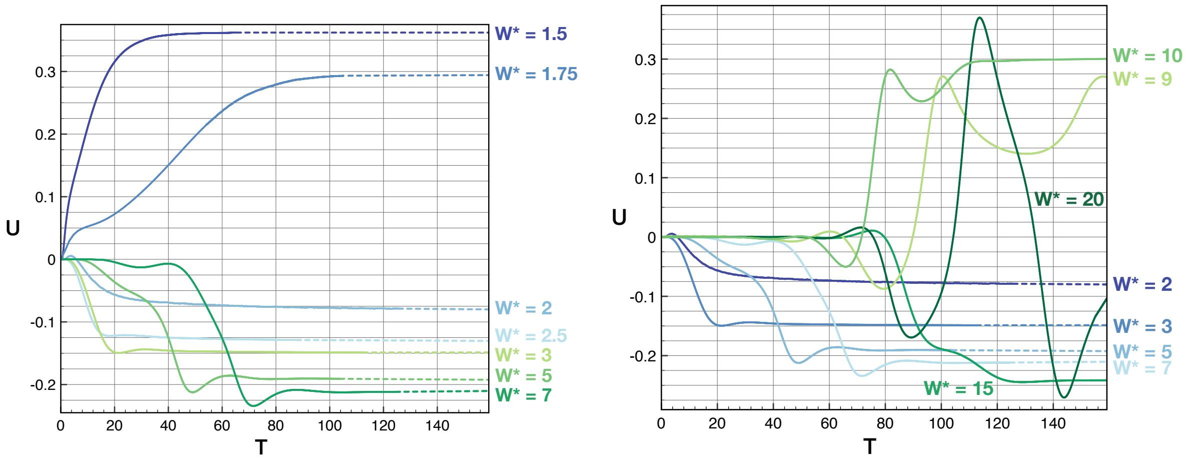


Figure 11.4: Velocity of the counter-rotating cylinder pair at $Re_\Gamma = 100$ for a series of non-dimensional widths W^*. Simulations are stopped whenever the velocity does not change more than 0.03% in the last time unit, and extended with a dashed line to indicate the steady-state value.

$1.25 \leq W^* \leq 10$ we could always find a high enough Reynolds number for which the velocity of the cylinders was unstable.

11.4 Discussion

We discuss the steady locomotion of the cylinder pair in terms of the flow fields presented in the previous section. We also discuss the transition between upwards and downwards motion, and the transition from downwards/jet-like motion to unstable solutions.

11.4.1 Flow visualization

Figure 11.6 visualizes the velocity field of the cylinders at steady-state using a Line Integral Convolution (LIC, Cabral and Leedom 1993), which can be interpreted as a space-filling streamline plot. We show the LIC field both for $Re_\Gamma = 50$ and $Re_\Gamma = 150$ in the frame of reference of the body center of mass. The lower Reynolds number solution shows a coherent elliptical region of circulating fluid surrounding the cylinder pair, traveling upwards with the cylinders, similar to the streamlines around a vortex pair moving at steady state. This is similar to the flow obtained past a stationary rotating cylinder pair with suppressed wake (Chan et al. 2011). In contrast, at the higher Reynolds number, there is a closed region of fluid around each cylinder. The incoming flow is distributed around and in between the cylinders, reminiscent of a jet-like flow where the fluid is ejected

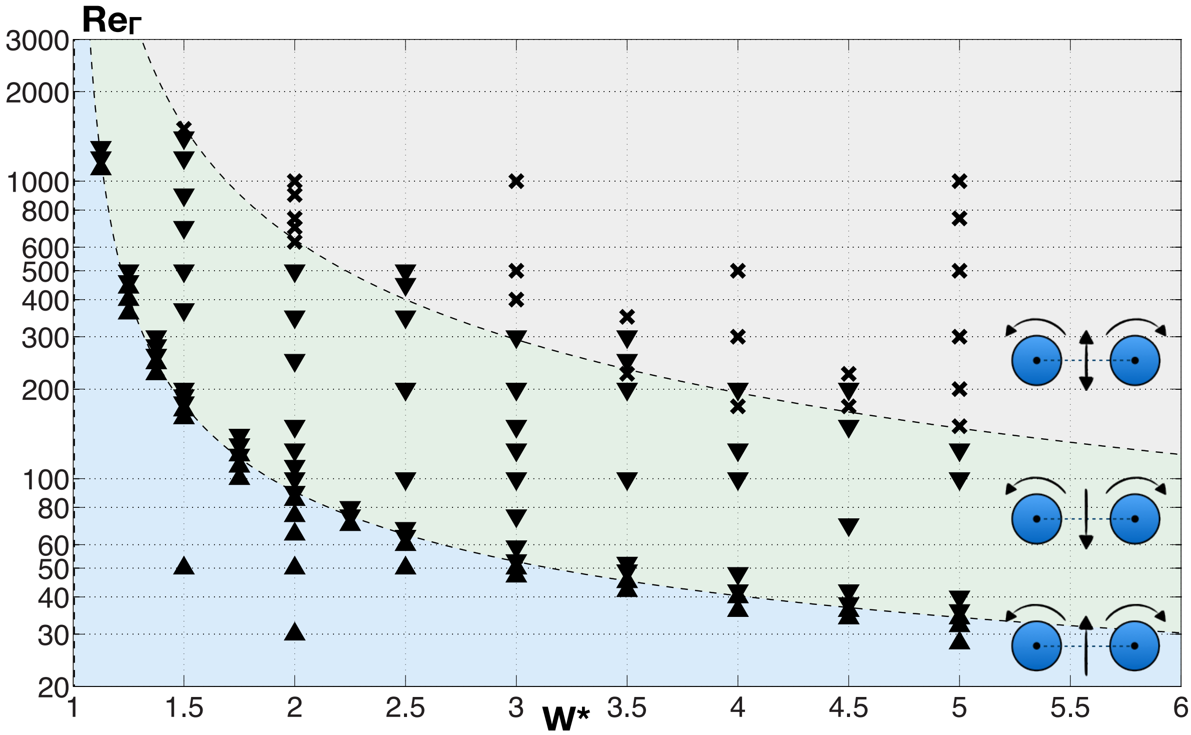


Figure 11.5: Transition between upward, vortex pair-like motion (upwards pointing triangles) and downward, jet-like motion (downwards pointing triangles) as a function of $Re_\Gamma = \Gamma/\nu$ and $W^* = W/D$. The crosses denote unstable motion.

through the opening between the cylinders. These results are consistent with (Sungnul and Moshkin 2009; Suwannasri 2013).

We also examine the two flow regimes discussed above by seeding the flow with tracer particles that are passively advected with the local velocity field. The evolution of the particles is shown in figure 11.7. The particles confirm the observations made based on the streamlines: for the lower Reynolds number, corresponding to upwards motion, the surrounding flow does not penetrate the closed region of fluid surrounding the cylinder pair, instead it is being deflected around it. At higher Reynolds number, the particles are pushed between the two cylinders in a jet-type flow.

11.4.2 Low Reynolds number regime

To compare the lower Reynolds number flow regime to the idealized case of a point vortex pair, we consider the velocity of the vortex pair as

$$U = \frac{u}{U_\theta} = \frac{\Gamma/(2\pi W)}{U_\theta} = \frac{1}{2W^*},$$

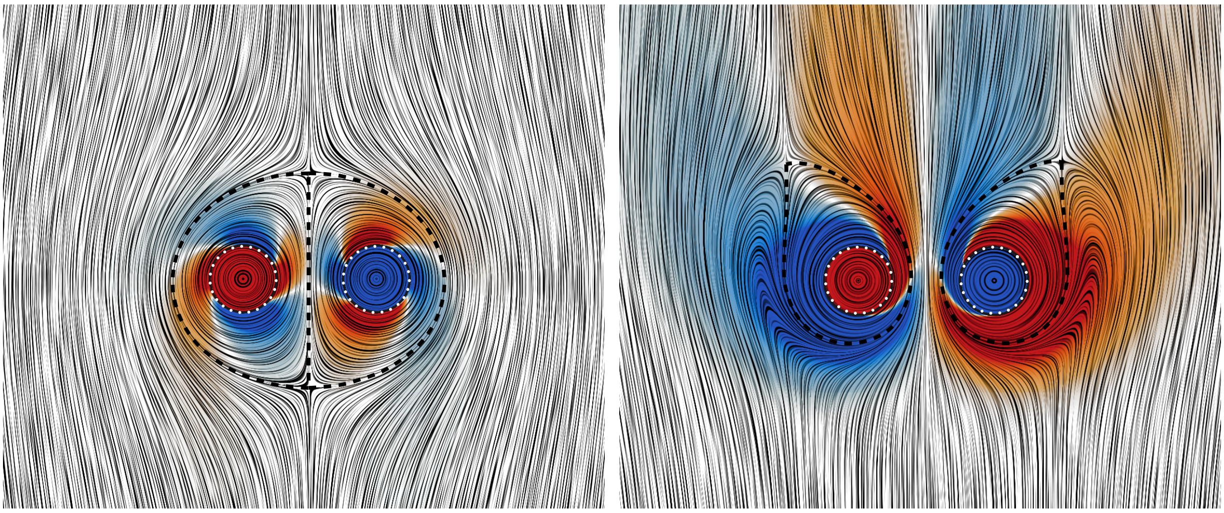


Figure 11.6: Line-integral convolution field (Cabral and Leedom 1993) illustrating the velocity field around the cylinders at Reynolds number 50 (left) and 150 (right), computed in the moving frame of reference of the cylinders. The cylinders are marked by dotted white lines, closed streamlines around the cylinders by dashed black lines and the colors refer to the vorticity field.

where we expressed the circulation Γ through the corresponding surface velocity for a cylinder of diameter D. In figure 11.8 (left) we plot the velocity of the cylinder pair at $Re_\Gamma = 10$ as a function of W^*, and observe that the steady-state velocity has an inversely linear relation to W^*. The right plot of figure 11.8 indeed confirms that the steady-state velocity at various Reynolds numbers below the transition is inversely proportional to the non-dimensional width, similar to a vortex pair. Linear fitting the data points gives prefactors of $(0.41, 0.45, 0.485, 0.518)$ for Reynolds numbers $(5, 10, 20, 30)$, very close to the inviscid value of 0.5.

11.4.3 Flow instability

The transition to unstable motion is shown in more detail in figure 11.9, for $W^* = 3$ and $W^* = 5$ at a variety of Reynolds numbers. For $W^* = 3$, starting at $Re_\Gamma = 125$ the cylinders move downwards in stable manner. As the Reynolds number increases, the cylinders' initial acceleration decreases causing an accumulation of vorticity near the surface. This shear layer develops an instability that grows as it is convected along the surface. Eventually the instability causes the boundary layer vorticity to detach and as it is shed into the wake, the cylinders accelerate in opposite direction. With increasing Reynolds number the momentum of the fluid shed into the wake increases, so that the cylinder's are gradually more perturbed by this process. This is visible in the $W^* = 3$ plot as growing oscillations

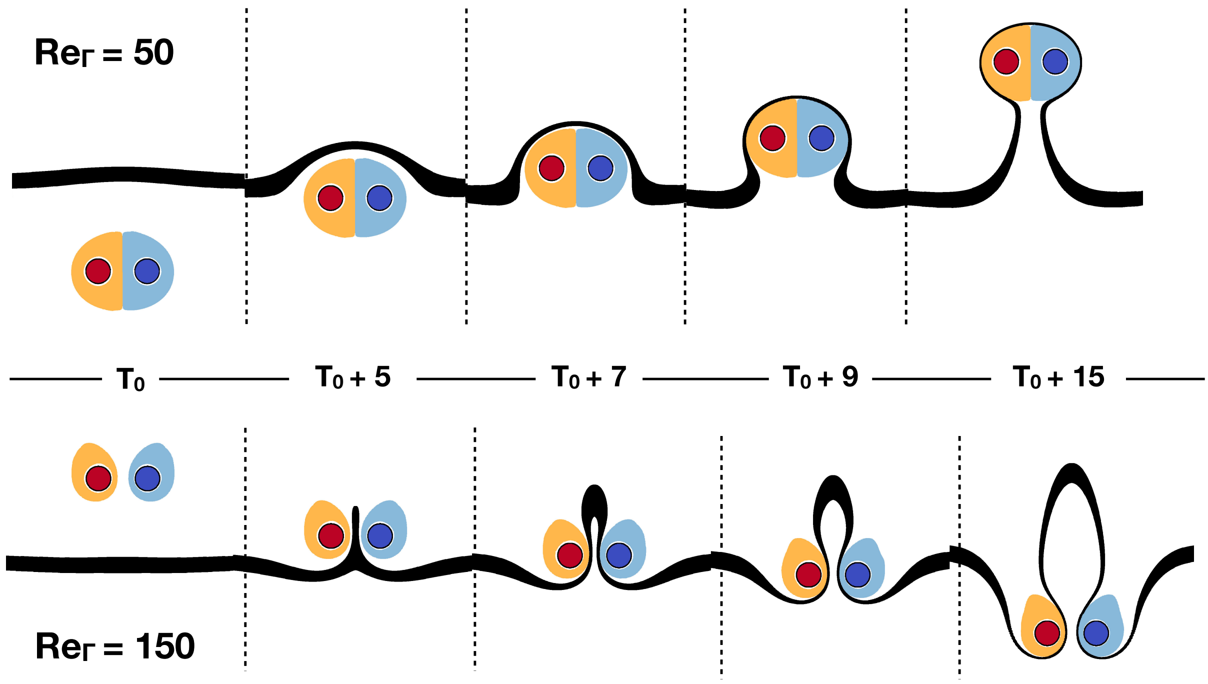


Figure 11.7: Passive tracer particles are advected with the velocity field around the cylinders at $Re_\Gamma = 50$ (top) and $Re_\Gamma = 150$ (bottom). The cylinders move at steady-state and are depicted by the dark red and dark blue circles. We seed passive tracer particles around the counter-clockwise rotating cylinder (in light red), around the clockwise rotating cylinder (in light blue) and in a strip upstream of the cylinder (in black). Animations are provided in the supplementary material.

amplitudes between $T = 25$ and $T = 50$ for $Re_\Gamma = 150$ up to $Re_\Gamma = 300$. For all these cases, however, the cylinder manages to restore to a steady velocity in downward direction. At $Re_\Gamma = 500$ this capacity is lost and the cylinders behave unpredictably. For such high Reynolds numbers the growth rate of the instability is too large for the cylinders to settle at a steady-state. Since the trigger for the initial perturbation in our setup is caused by non-reproducible floating-point arithmetic, repeated simulations in this unstable regime can give drastically different outcomes.

The cylinder pair with $W^* = 5$ has a qualitatively similar behavior, in this case the cylinders become unstable at $Re_\Gamma = 200$. Surprisingly, though, at $Re_\Gamma = 300$ the pair has recovered a seemingly stable solution where the motion direction is upwards, and vorticity is shed continuously into the wake. Then at $Re_\Gamma = 500$ the cylinders initially are unstable until the velocity finally keeps hovering around a negative value. The LIC field of the latter two cases are shown in figure 11.10. The velocity field at at $Re_\Gamma = 500$ is comparable with that of the low Reynolds number case in figure 11.6, where both cylinders move downwards.

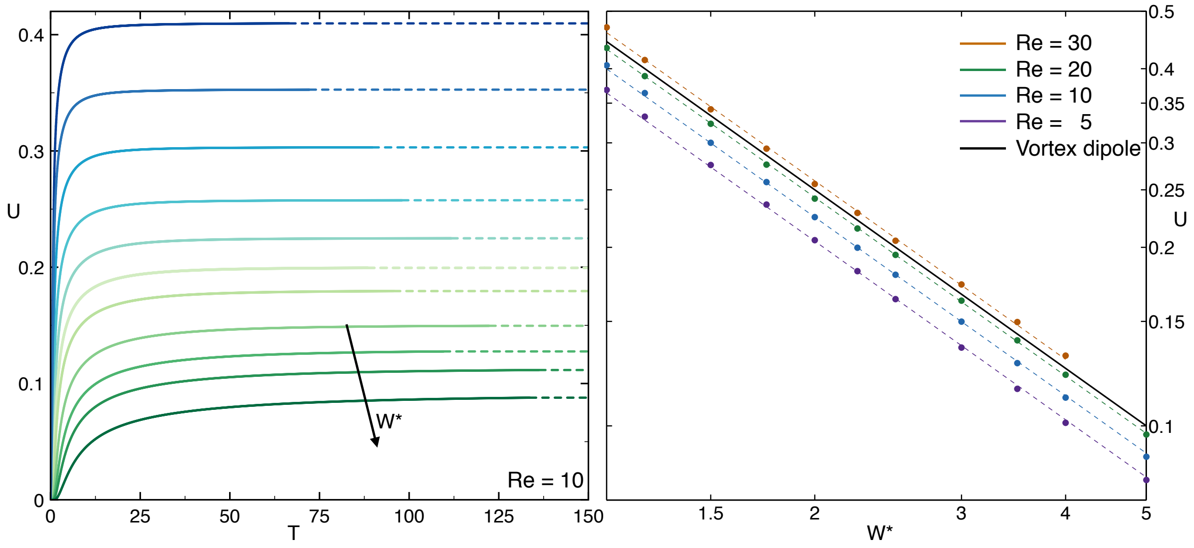


Figure 11.8: On the left, velocity of the cylinder pair at $Re_\Gamma = 10$ for a series of W^*. From blue to green: $W^* = [1.125, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 3.0, 3.5, 4.0, 5.0]$. On the right, a logarithmic plot of the steady-state velocity against the distance W^* between the cylinders for various Re_Γ compared to the velocity for a ideal vortex pair (black).

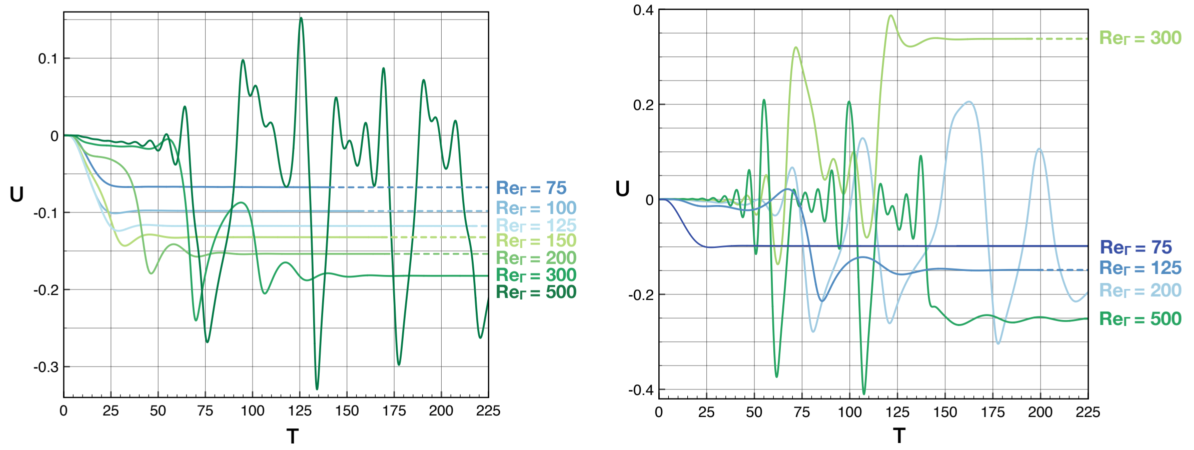


Figure 11.9: Vertical velocity of the cylinders with $W^* = 3$ (left) and $W^* = 5$ for Reynolds numbers around the instability.

However at $Re_\Gamma = 300$ the flow is very different from the previous upward solution. The closed elliptical region of flow surrounding the cylinder pair is absent, instead the flow travels in between the cylinders and the largest flow velocities occur at the outer sides of the pair.

Interestingly, a quasi-periodic case was found for the evolution of $W^* = 3$ and $Re_\Gamma = 1000$,

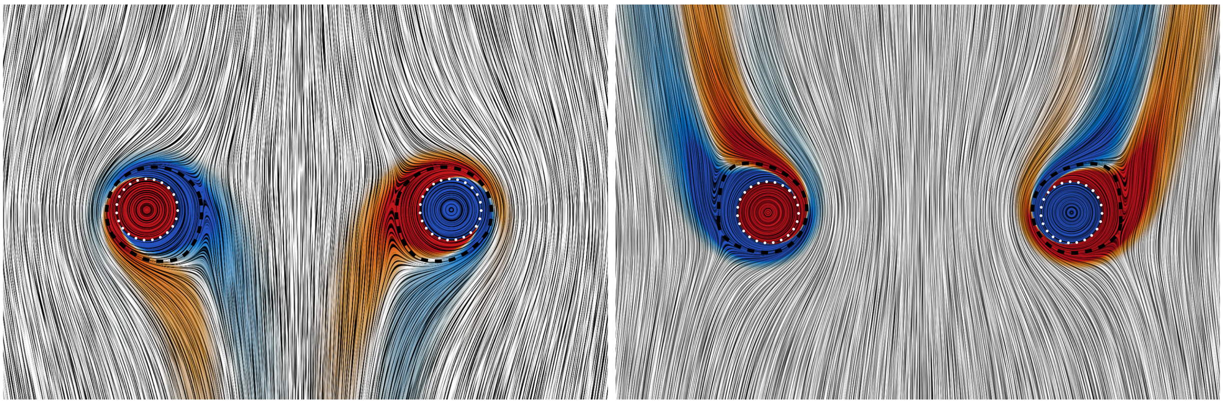


Figure 11.10: Line-integral convolution field (Cabral and Leedom 1993) of the velocity around the cylinders with $W^* = 5$ at $Re_\Gamma = 300$ (left) and $Re_\Gamma = 500$ (right), computed in the moving frame of reference of the cylinders. The cylinders are marked by dotted white lines, closed streamlines around the cylinders by dashed black lines and the colors refer to the vorticity field.

as shown in figure 11.11. These cylinders show a quasi-periodic vortex shedding that propels them forward incrementally. After a vortex pair has been shed, the boundary-layer instability builds up afresh until it finally sheds. With this pattern the cylinders manage to move around 40 diameters during 300 revolutions.

11.4.4 Transient behavior

Whereas above we have always started the cylinders from rest, here we consider the case of a change in circulation after the cylinder pair has reached steady state in either the upwards or downwards moving regime. Specifically, we change the circulation magnitude of both cylinders according to

$$\Gamma(T) = \begin{cases} \Gamma_0, & T \leq T_0 \\ \Gamma_0\left(1 + \frac{1}{2}\left(\frac{\Gamma_1}{\Gamma_0} - 1\right)\left[1 - \cos\left(\pi \frac{T - T_0}{T_1 - T_0}\right)\right]\right), & T_0 < T < T_1 \\ \Gamma_1, & T \geq T_1 \end{cases} \tag{11.1}$$

where the non-dimensionalization is based on the state values at $T \leq T_0$. We use $T_0 = 200$ and $T_1 = 210$ to smoothly change the circulation from Γ_0 to Γ_1.

In figure 11.12 we show what happens when the cylinder pair is subjected to an increase in circulation when starting from a vortex pair-like regime (left), and to a decrease in circulation when starting from a jet-like regime (right). In the former case, we see that the vortex pair-like regime can be extended to much higher Reynolds numbers when

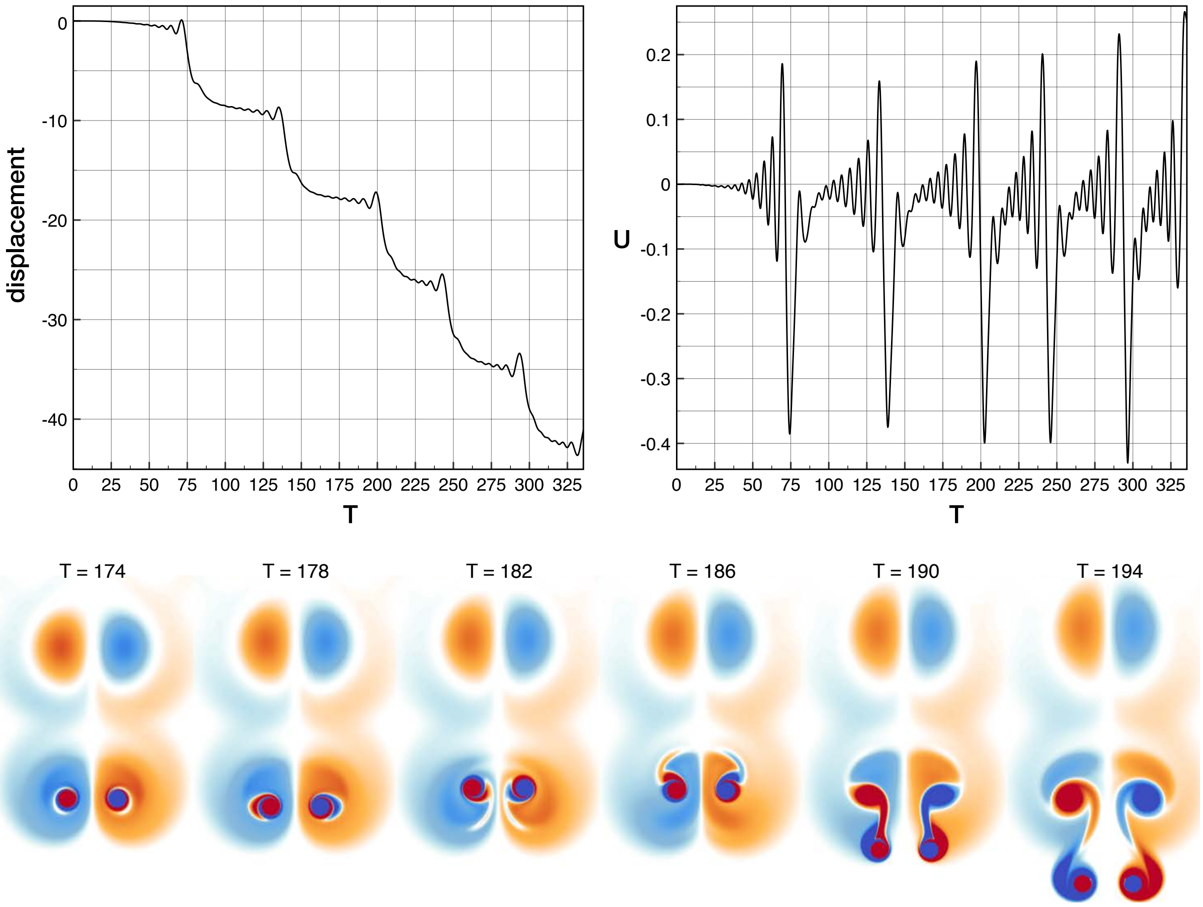


Figure 11.11: Displacement (top left, non-dimensionalized with the diameter D), (top right) and vorticity field at selected times (bottom) for the cylinders with $W^* = 3$ and $Re_\Gamma = 1000$. Further animations are provided in the supplementary material.

starting from a steady-state solution. At $Re_\Gamma = 120$ we observe that during the transition, the cylinder pair initially slows down, yet after the new circulation value is reached it speeds up again. The steady-state at $Re_\Gamma = 120$ corresponds to a 40% larger velocity than the steady-state at $Re_\Gamma = 80$.

11.5 Conclusions

In this work we discussed a simple mechanism for self-propulsion in a viscous fluid consisting of a steadily counter-rotating cylinder pair. For low rotation rates the cylinders behave as an ideal vortex pair, moving in the direction of the internal surface velocity. The flow is characterized by an elliptical region of fluid surrounding the cylinders enclosed by a bounding streamline, around which the external fluid is deflected. We demonstrated

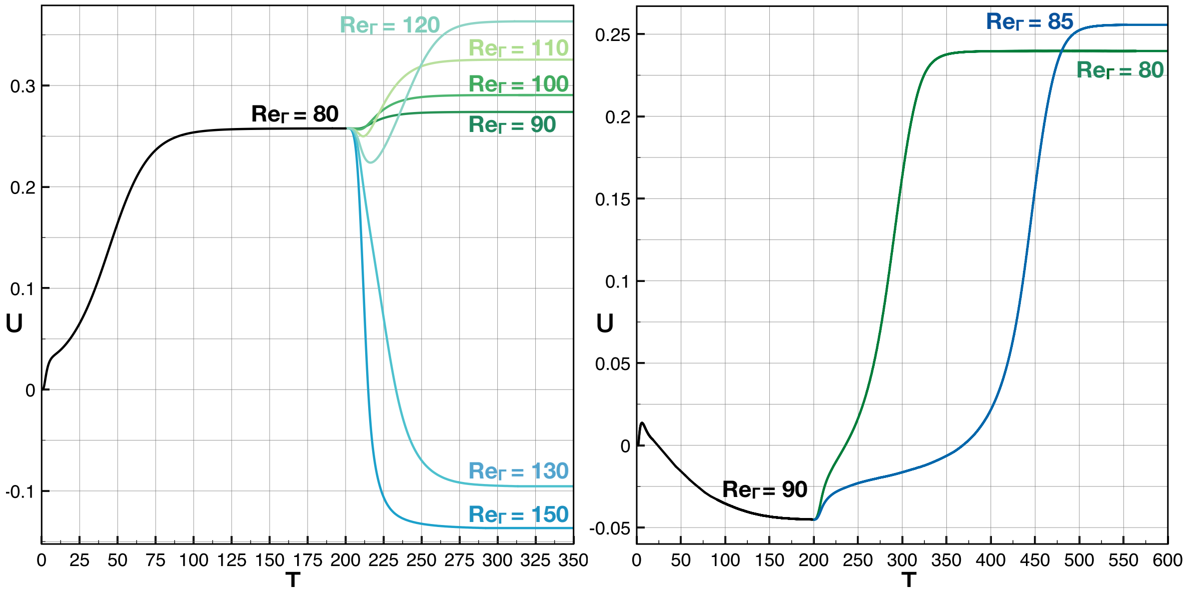


Figure 11.12: Cylinder pair behavior in response to an increase in circulation starting from $Re_\Gamma = 80$ (left), and to a decrease in circulation starting from $Re_\Gamma = 90$ (right). The width is $W^* = 2$. Animations of the flow evolutions for selected cases are provided in the supplementary material.

that in this regime the velocity of the cylinders is inversely proportional to the width, as for the ideal vortex pair. Increasing the rotation rate shows a transition to a new state where the cylinders move in the opposite direction. In this regime the external flow is accelerated in between the cylinders and expelled like a jet. The Reynolds number of transition between these two states depends on the width between the cylinders with an inverted exponential relationship, so that larger widths requires smaller rotation rates to move from the vortex-pair-type to the jet-type flow.

Increasing the rotation rate even further shows an unstable regime as the instabilities growing in the boundary layer of each cylinder are shed into the wake. For small rotation rate increments, the cylinders can recover from the first shedding event to settle in a steadily-moving jet-like state. At higher rotation rates their motion is dominated by discrete vortex shedding events, each of which may result in upwards or downwards motion depending on the direction in which the vorticity is shed. This generally results in unsteady, irregular behavior but we demonstrate that some isolated stable or quasi-periodic solutions exist within the unstable regime that are sustained for longer periods of time.

We believe this simplified form of locomotion can serve as inspiration for the development of engineering devices in fields such as robotics and fluid mixing. Changing the rotation of one of the cylinders can enable the system to rotate, a useful property for robotic devices. Future work will investigate the energy efficiency and performance when the cylinders' rotation has an oscillatory character. Furthermore we will consider the extension of this system to three dimensions. On the latter subject one can imagine a variety of geometric configurations, ranging from simply a pair of rotating spheres, to planar triangular configurations with rotating spheres on the vertices, or rotating cylinders on the edges of morphing structures, a subject of ongoing investigations in our lab.

Acknowledgments This work was performed together with Guido Novati as part of his M.Sc. degree at Delft University of Technology.

CHAPTER 12

Multiresolution 3D flow simulations

In this chapter we present the capabilities of the 3D multiresolution flow solver as presented in chapters 9 and 10. For validation we study the flow past a sphere and the evolution of a self-propelled swimmer. We compare with established results from literature as well as results and performance of our uniform resolution flow solver. Lastly we show a simulation of several side-by-side 3D self-propelled swimmers.

All of the simulations were performed on the Cray XC30 'Piz Daint'. Each compute node of this machine contains 8 Intel Xeon E5-2670 CPUs with a clock rate of 2.6 GHz, and an NVIDIA Tesla K20X GPU. One compute node has a memory of 32 GB and theoretical peak performances in single precision of 332.8 GFLOP/s for the CPUs and 3.95 TFLOP/s for the GPU. The MRAG multiresolution code was compiled with version 4.8.2 of the GNU C++ compiler, whereas the PPM uniform resolution code was compiled with version 14.0.1 of the Intel Fortran compiler. The results of this chapter are obtained with single precision arithmetic, and all simulations employed four nodes in the master/slave configuration as described in section 10.5. Unless otherwise mentioned we use penalization parameter $\lambda = 10^4$, refinement and compression thresholds $\epsilon_R = 10^{-3}$, $\epsilon_C = 10^{-5}$, multipole settings $\theta = 0.5$ and $p = 6$, and adaptation frequency $N_{\text{adapt}} = 10$.

12.1 Flow past a sphere

In this section we consider a flow with $\mathbf{U}_\infty = (U_\infty, 0, 0)^T$ past a fixed sphere with diameter D in a fluid with kinematic viscosity ν. Up to $Re = U_\infty D/\nu \approx 212$ the flow is axisymmetric and stable (Tomboulides and Orszag 2000), and the detailed characteristics of the steady-state flow are well documented (T. A. Johnson and Patel 1999). We use these reference results to validate our method for Reynolds numbers 50, 100, 150 and 200. Our simulations

Re	C_D	Θ	x_c	y_c	x_s
50	1.573 (1.571)	139.3 (139.2)	0.637 (0.640)	0.211 (0.212)	0.409 (0.420)
100	1.096 (1.084)	126.7 (126.6)	0.758 (0.761)	0.289 (0.292)	0.872 (0.887)
150	0.904 (0.892)	120.6 (121.0)	0.813 (0.831)	0.326 (0.331)	1.133 (1.216)
200	0.794 (0.778)	116.5 (117.2)	0.838 (0.885)	0.350 (0.361)	1.263 (1.461)

Table 12.1: Diagnostics for axisymmetric flow past a sphere compared with values of T. A. Johnson and Patel (1999) in parentheses. See figure 12.1 for the definition of the geometric parameters.

are performed with effective resolution 4096^3 and a sphere radius of 0.025, resulting in a resolution of approximately 100 grid points across the radius. We set the time step criterion LCFL $= 0.01$. The sphere center is placed at $x = 0.15$ and $y = z = 0.5$ in a unit cube domain with constant inflow velocity $U_\infty = 0.1$. For the mollification of the characteristic function we use the FDMH$_1$ method of Towers (2009).

In table 12.1 we compare the drag coefficient at steady state as well as the separation angle, separation bubble length and coordinates of the center of the separation vortex with the reference data of T. A. Johnson and Patel (1999). The drag coefficient is defined as $C_D = 8F_D/(\pi U_\infty^2 D^2)$, where F_D is the drag force, and a visual definition of the geometric metrics is given in figure 12.1. For the late time results we observed a slight loss in vertical symmetry, possibly due to the relatively high refinement and compression values. Nevertheless, our results for drag and flow characteristics agree well with the reference.

We take the simulation of the sphere at $Re = 50$ and analyze the performance details of the multiresolution solver. Figure 12.2 shows how the time of a simulation is distributed. The velocity solver takes by far most of the computational time, and we will detail this in the next paragraph. The second costliest operation is diffusion, which, at this low Reynolds number, is performed with Local Time Stepping (LTS) as explained in section 9.2. Using LTS means one global time step entails many more updates on the finer levels of resolution, which explains its large cost relative to the other right-hand side computations. Compression and refinement are expensive per operation, but because they are only performed every ten time steps, their contribution to the accumulated timings is small. Particle advection is more expensive than standard finite-difference operations due to the irregular memory access patterns and tensorial support of the multidimensional interpolation kernels. Stretching and penalization rely solely on applying a finite-difference scheme and are therefore the least expensive operations. For this case, the characteristic function of the obstacle can be computed analytically and the obstacle is fixed in place, so that those operations have a negligible impact on the simulation time.

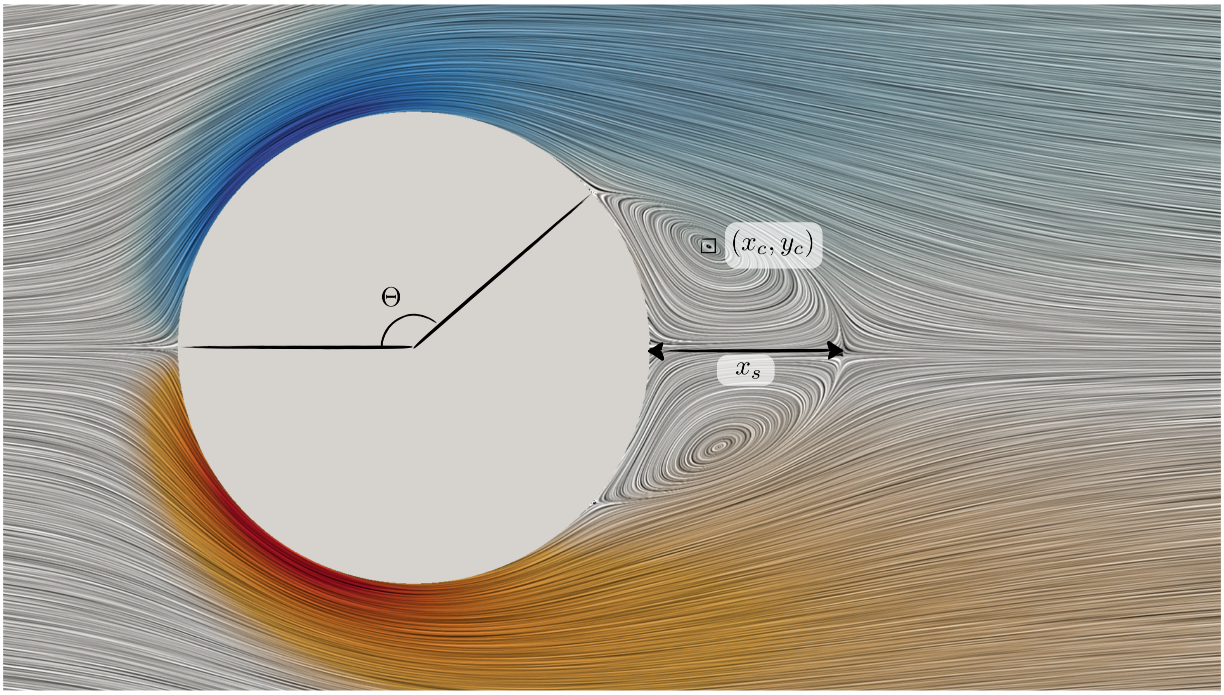


Figure 12.1: Definitions of the vortex center (x_c, y_c), the separation length (x_s) and the separation angle Θ illustrated on the streamlines of the flow past a sphere at $Re = 50$. The colors correspond to the out-of-plane vorticity component.

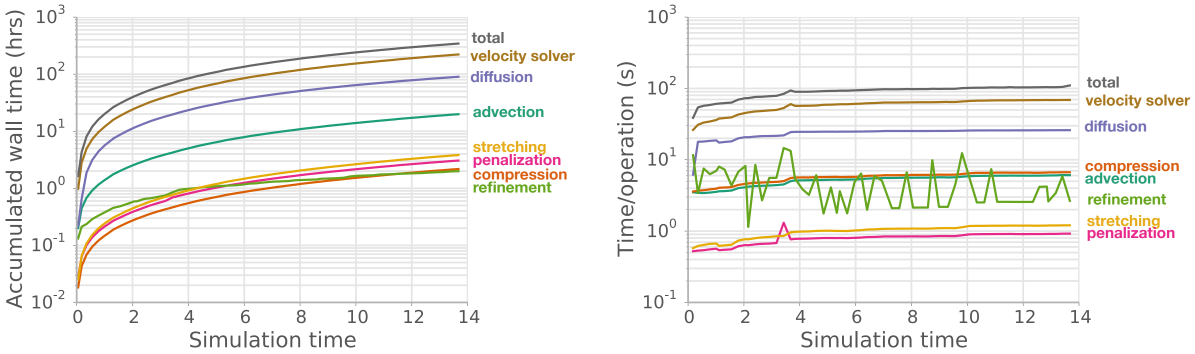


Figure 12.2: Distribution of time spent within the multiresolution solver for the flow past a sphere at $Re = 50$, accumulated (left) and per operation (right), as a function of the simulation time T.

A closer look to the time distribution within the velocity solver is shown in figure 12.3, where we show the average timings per component of the velocity field measured from the master rank. The largest contribution corresponds to the combined evaluation of the direct and indirect interactions, and it dominates the time spent in the solver. The next paragraph details this further. Communication of the interaction lists and source arrays

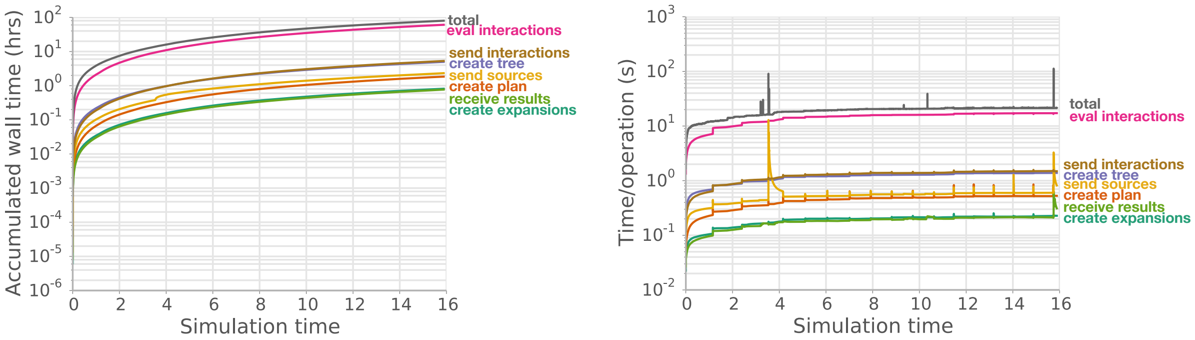


Figure 12.3: Distribution of time spent within the hybrid CPU/GPU velocity solver, accumulated (left) and per operation (right), for the flow past a sphere at $Re = 50$.

between the master and slave nodes is non-negligible but could be improved by performing non-blocking operations and coarser granularity, as explained in section 10.5. Receiving the results involves each node accumulating their CPU and GPU results, and subsequently sending it back to the master, and these operations takes a negligible amount of time. Constructing the tree, the plan (i.e. interaction lists) and the multipole expansions are all inexpensive operations compared with the evaluation phase.

The distribution between the evaluation of the direct and indirect interactions is shown in figure 12.4 in terms of number of interactions (left) and wall-clock time (right). The number of direct (particle-brick) interactions is about 25 times larger than the number of indirect (multipole-brick) interactions. Nevertheless the time spent in the indirect interactions dominates, because the direct interactions are evaluated on the GPU. This analysis shows that developmental work towards optimizing the time spent in computing indirect interactions would be most effective in reducing the time-to-solution.

12.2 Single self-propelled swimmer

The shape and motions of a deforming swimmer are implemented using the semi-analytical equations presented in appendix B. We use the Reynolds number $Re = (L^2/T)/\nu$ based the swimmers' length L and swimming frequency T. The midline kinematics are described by the anguilliform motion pattern used throughout this work, where the lateral displacement of the midline $y_s(s,t)$ in a local frame of reference is given as

$$y_s(s,t) = 0.125L\frac{0.03125 + s/L}{1.03125}\sin\left[2\pi\left(\frac{s}{L} - \frac{t}{T} + \phi\right)\right], \tag{12.1}$$

where we set the phase shift $\phi = 0$ for the single swimmer. The swimming motion is ramped-up during the first swimming cycle through multiplication with a quarter-period

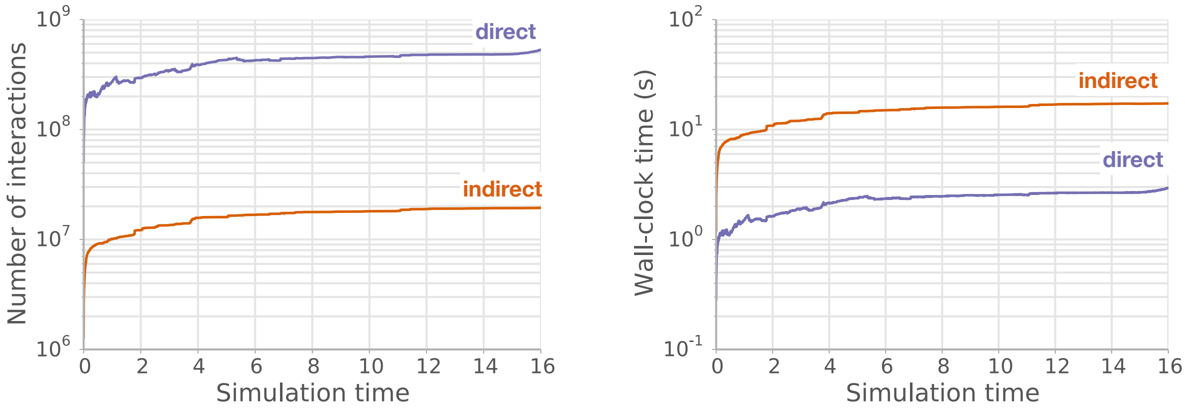


Figure 12.4: Number of direct and indirect interactions (per target-brick, left) and time distribution (right) for the velocity evaluation when simulating the flow past a sphere at $Re = 50$.

sine wave. The three-dimensional morphology is chosen from Kern and Koumoutsakos (2006), in turn based on Carling, Williams, and Bowtell (1998). The cross-sections of the shape are elliptical and the width $w(s)$ and height $h(s)$ functions are given as

$$w(s) = \begin{cases} \sqrt{2w_h s - s^2} & 0 \le s < s_b, \\ w_h - (w_h - w_t)\left(\dfrac{s - s_b}{s_t - s_b}\right)^2 & s_b \le s < s_t, \\ w_t\left(\dfrac{L - s}{L - s_t}\right) & s_t \le s \le L, \end{cases} \tag{12.2}$$

and

$$h(s) = b\sqrt{1 - \left(\frac{s - a}{a}\right)^2}, \tag{12.3}$$

where $s_b = w_h = 0.04L$, $s_t = 0.95L$, $w_t = 0.01L$, $a = 0.51L$ and $b = 0.08L$. The swimmer's geometry is mollified using the smooth function of section 5.2.

For Reynolds number $Re = (L^2/T)/\nu = 550$ we have no reference results in literature and therefore we compare with our own uniform resolution code that was used in part II of this thesis. For the multiresolution results, we set the fish length $L = 0.25$ and swimming frequency $T = 1$ in a unit cube domain. Initially the swimmer's center of mass is located at $x = 0.65$ and $y = z = 0.5$. Multiresolution results are presented with an effective resolution of 1024^3, resulting in 256 grid points along L, and LCFL $= 0.01$.

The uniform resolution code employs the same temporal and spatial resolution of the swimmer, and exploits the dynamic domain growth capacities of PPM library described in section 2.3. The uniform resolution code is executed on 48 compute nodes. Figure 12.5 shows that forward, lateral and angular velocities of the two simulations are comparable.

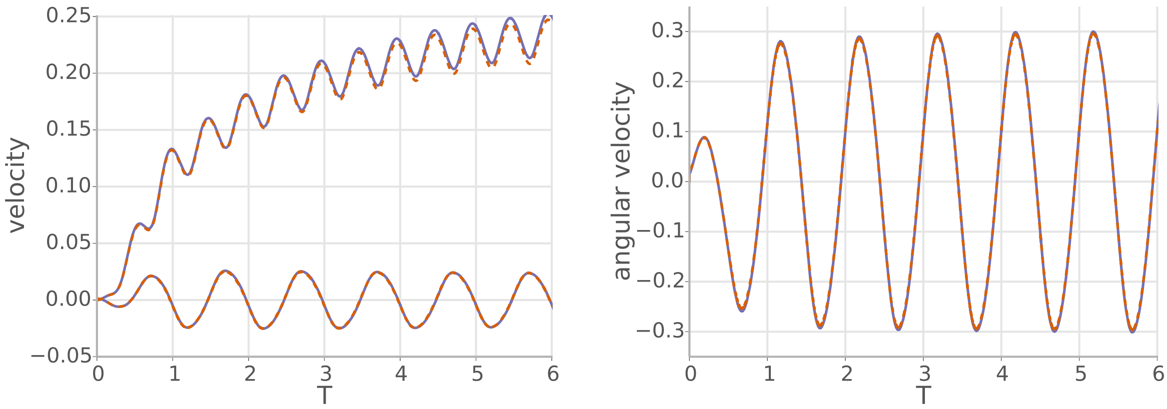


Figure 12.5: Forward and lateral velocity (left) and angular velocity (right) of a self-propelled swimmer at Re=550. Multiresolution MRAG results (solid blue) compared with uniform resolution PPM results (dashed red).

A performance comparison between the uniform resolution PPM code and the multiresolution MRAG code is given in figure 12.7. The left plot shows that the number of computational elements is similar, due to the dynamic domain growth of the PPM code. In contrast, the MRAG code simulates a unit box with effective resolution of 1024^3, and the wavelet-based adaptation gives a compression rate of 90× at $T = 6$. The plot highlights the effectiveness of domain growth according to the vorticity support in PPM, which is something desirable for the MRAG code as well. However we note that at this low Reynolds number, the absence of strong gradients and large scale-separations does not benefit wavelet-based grid-adaptation with respect to uniform resolution codes. This can be seen in the grid adaptation at $T = 6$ shown in figure 12.6, where the swimmer and the wake are essentially covered by a locally uniform resolution grid adapted to the support of the vorticity.

The right plot in figure 12.7 shows the wall-clock time of both simulations. Since the uniform resolution simulation runs on 48 nodes, compared with 4 nodes for the multiresolution simulation, we scaled up the timings of the uniform simulation by a factor of 12 (dashed line). This is an idealized assumption since we can not expect perfect scaling. Furthermore the memory footprint of 4 nodes would not be enough to support this simulation using PPM. With this in mind, the required time is slightly lower for PPM than for MRAG. This can be attributed due to the additional overhead of the multiresolution framework, which provides no significant reduction in active computational elements for this simulation. The performance is expected to improve for higher Reynolds number flows, where the larger scale-separation can be more effectively exploited using wavelet-based adaptivity. Nevertheless we note that the multiresolution framework brings full 3D FSI

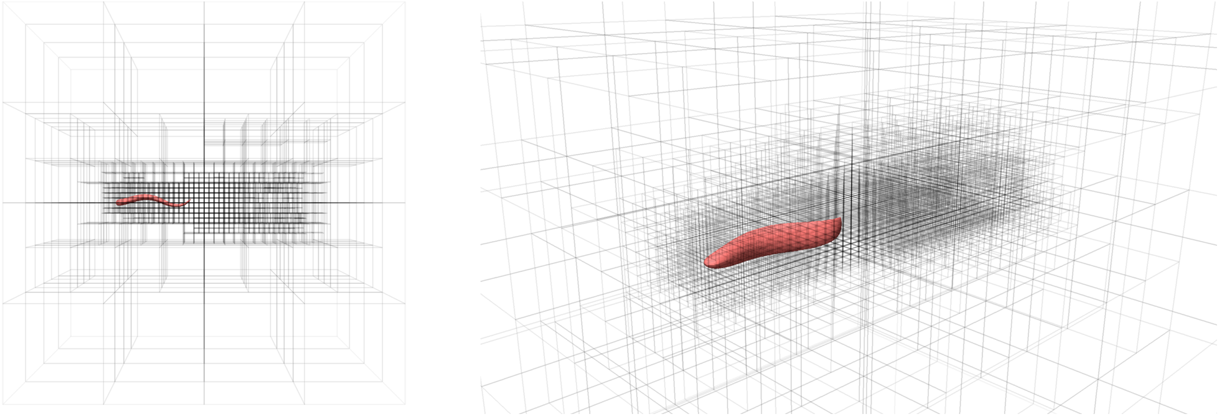


Figure 12.6: Grid blocks in the domain of the $Re = 550$ multiresolution swimmer simulation at $T = 6$ from a top view (left) and perspective view (right). The swimmer is colored in red.

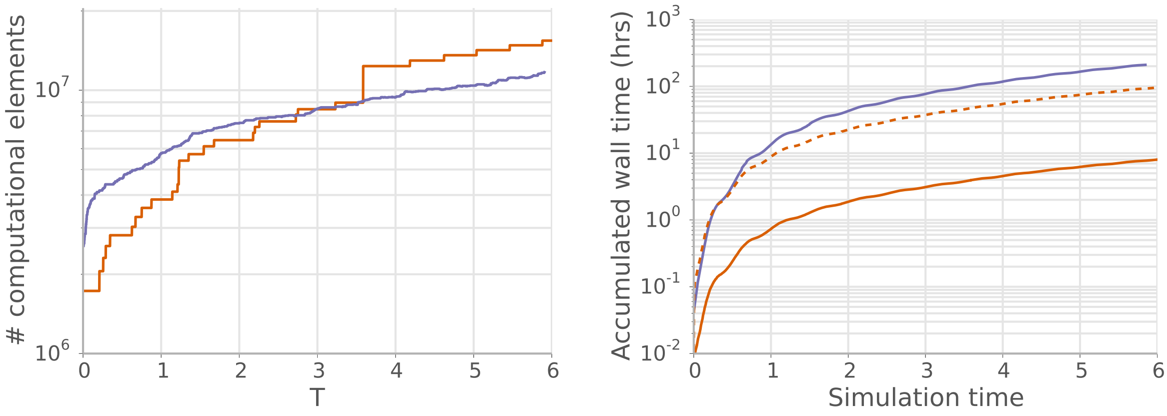


Figure 12.7: The left plot shows the number of computational elements of the uniform resolution code with dynamic domain adaptation (red) and the multiresolution code (blue). The right plot shows the wall time for the same two simulations. The dashed red line corresponds to the uniform resolution timings were it run on the same number of compute nodes as the multiresolution code, under idealized conditions.

simulations within reach of computer systems with only a few nodes, rather than requiring a supercomputing environment.

12.3 Four self-propelled swimmers

To show the flexibility and capabilities of the multiresolution method we present the results of four side-by-side self-propelled swimmers with constant $L = 0.125$, $T = 1$ and

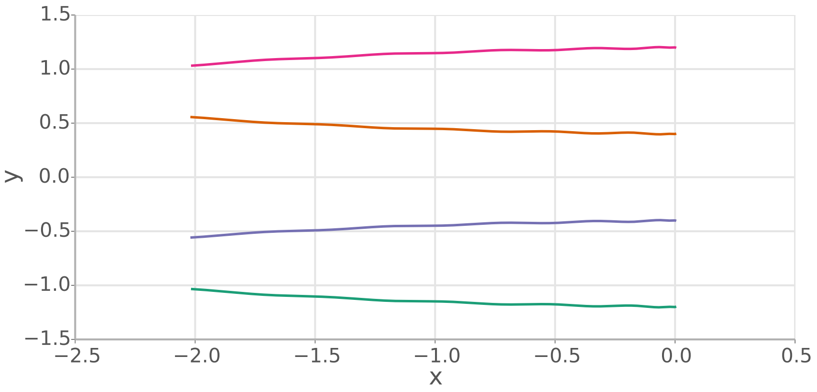


Figure 12.8: Center-of-mass trajectory of four parallel out-of-phase swimmers at $Re = 7143$, normalized by their length. The swimmers move from right to left.

$Re = 7143$. The motion of the swimmers is given by equation (12.1), where the difference in ϕ between neighboring swimmers is 1/2. Their initial positions are $x = 0.75$ and $y = (0.35, 0.45, 0.55, 0.65)$. We set LCFL $= 0.1$ and use an effective resolution of 2048^3. All other settings are the same as in the previous section.

Figure 12.8 shows the center-of-mass trajectory of each of the swimmers, normalized by their lengths. The middle pair of swimmers move apart, whereas the two outer pairs converge. The trajectories of the swimmers that share the same phase shift are almost identical, and the forward velocities of all four swimmers differ by less than 1% at $T = 5$. This indicates that the early behavior is mostly determined by the asymmetry in the initial ramp-up, rather than fluid-mediated interactions. Consistent with Gazzola, Chatelain, et al. (2011), the fluid mediated interactions at later times will likely cause the outer two swimmers to diverge while the middle pair would initially converge, then diverge as they enter each other's vicinity.

The vorticity field evolution during the simulation is shown in figure 12.9. The vortex rings produced by the swimmers at the end of each stroke collide head-on, reminding of the classical experiments of Oshima (1978). The vortex rings stretch and grow in diameter until their vorticity has diffused. Except for the rings produced during the first stroke, the downstream end of the vortex rings is stronger than the upstream end. This is consistent with our lower Reynolds number observations for anguilliform swimming in chapter 7.

The vortex ring collision means that the individual rings do not diverge from the swimmer's trajectory, instead creating a series of vortical structures along the symmetry plane between two adjacent swimmers. This enables subsequent vortex rings to interact with the upstream part of the previous ring, resulting in complex dynamics as shown in figure 12.10. The

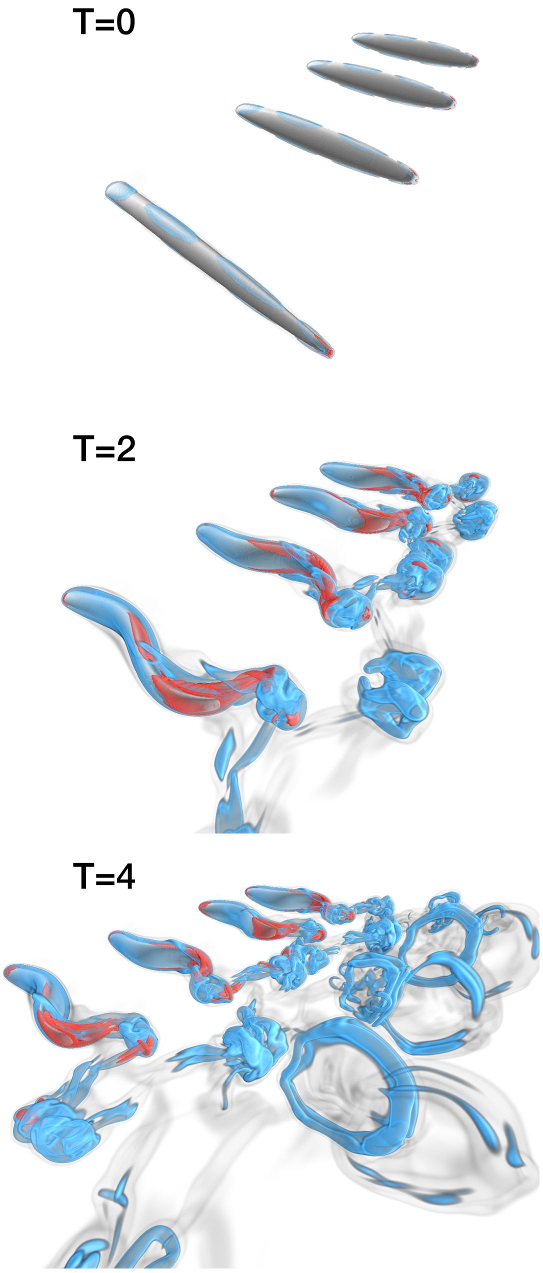


Figure 12.9: Vorticity field of four parallel self-propelled swimmers at indicated times. Strong vorticity is indicated with red, intermediate with blue and weak with light gray.

downstream leg of the incoming vortex ring wraps around the existing structure, while several other instabilities on the ring develop and distort its shape.

These simulations highlight for the first time the complexity of the vortex dynamics involved in multiple 3D self-propelled swimmers. The multiresolution software allows a user to specify an arbitrary array of swimmers, including vertical stacking to create an

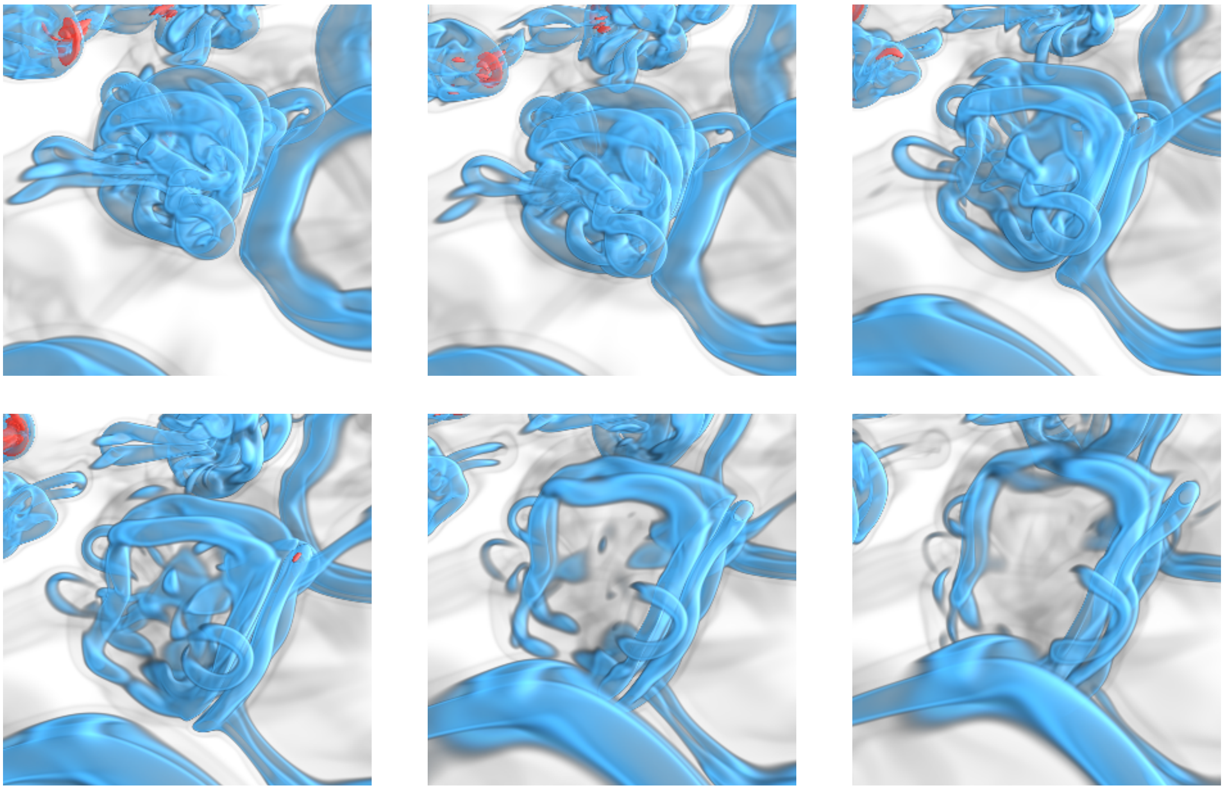


Figure 12.10: Zoom-in of the vortex dynamics in the midplane of the simulation between $T = 3.5$ and $T = 4.5$.

entire school. These capabilities can be exploited to assess the impact of the fluid-mediated interactions involved in schooling on the individual swimmers' trajectories, speed and efficiency.

Acknowledgments This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project s436. Figures 12.9 and 12.10 were rendered using the VolumePerception software (Rossinelli 2011).

Part IV

Conclusions

CHAPTER 13

Summary and perspectives

In this work we have studied a range of 3D incompressible flow simulations with remeshed vortex methods (rVM). The first section of this chapter summarizes the results and reviews the contributions of each part of this thesis. The second section discusses future research directions and follow-up work.

13.1 Summary

Vortex dynamics We have established the accuracy of the rVM compared with traditional pseudo-spectral methods for 3D periodic vortical flows at two different Reynolds numbers. For the Taylor-Green vortex at $Re = 1600$ we showed that the rVM can accurately reproduce the reference pseudo-spectral results on a quantitative basis. For the vortex tube reconnection at $Re = 10^4$ a close agreement is found until the flow becomes unstable, after which qualitative correspondence is still achieved. The accuracy of the rVM is shown to be sensitive to the choice of the interpolation kernel, with the six-point support M_6^* kernel reaching an accuracy comparable to pseudo-spectral results.

The rVM was then employed to study the dynamics of vortex reconnection at Reynolds number $\mathrm{Re} = 10^4$. We deduced that during the first reconnection, the curling of vortex lines causes an axial flow in the vortex tubes that persists until a late-time second reconnection. We hypothesized that this axial flow is correlated to a $-7/3$ slope in the energy spectrum at late times. Indeed when adding an initial axial flow to the vortex tubes we found a $-7/3$ slope also during the first reconnection. The results also showed that the threads which appear during the first reconnection are subject to a Kelvin-Helmholtz instability that breaks up their structure. This instability is expected to be even more dominant at higher Reynolds numbers, casting doubt on the idea of a vortex reconnection cascade, at least in this isolated form.

Fluid-solid interaction We have extended the rVM to incorporate two-way fluid-solid interaction in a flexible and robust manner. This allows the coupling of the flow solver with an optimization algorithm, creating a unique reverse-engineering framework for 3D deforming obstacles in viscous fluids.

Using this framework we found that the C-start, the escape pattern employed by natural fish, is an optimal escape mechanism. Analysis of the results showed that the effectiveness of the C-start hinges on trapping and acceleration of large regions of fluid through strong midline curvatures. We also showed that the natural escape motion can be improved by increasing the curvature during the preparatory stroke, although at a large energetic expense.

We then optimized the shape of swimmers with anguilliform kinematics separately for speed and efficiency. The optimal shapes show several characteristics common to natural swimmers such as a caudal fin and streamlined profile. However, our optimal shapes outperform natural larval zebrafish by 40% for speed and 135% for efficiency. We highlighted the significance of morphological features in the optimal shapes, such as an inflection point in the posterior width profile and the presence of a caudal fin.

Finally we presented for the first time optimal morphokinematic combinations for swimmers that maximize either speed or efficiency. Our results show that optimally fast steady-state swimming is characterized by kinematics similar to a C-start. Our optimally fast solution swims almost three times faster than natural anguilliform zebrafish. Even with the zebrafish' shape, such kinematics are highly effective, although the corresponding large energy expenditure suggests that natural swimmers could only sustain this for short time. The optimally efficient swimmer is characterized by a large volume and its kinematics are similar to carangiform fish, although even with an anguilliform gait large efficiencies can be obtained. Exploration of the parameter landscape indicates that the fast solution is highly specialized and strongly dependent on the midline kinematics, whereas efficient solution is less sensitive and depends largely on the morphology.

Multiresolution flow simulations We presented a numerical software based on wavelet-adaptivity and the rVM to perform 3D multiresolution FSI simulations. Our solver is designed for a small memory footprint enabled by the high compression rate of wavelets, and aims to perform 3D direct numerical simulations on only few compute nodes. To this end we designed and implemented a hybrid CPU/GPU multipole-based velocity solver and showed its performance for practical flow applications.

We used an existing 2D rVM solver based on the same multiresolution principles to study the locomotion of a counter-rotating cylinder pair, showing how the behavior surprisingly transitions from dipole-like to jet-like as the rotational Reynolds number or distance between the cylinders increases.

The 3D solver was validated for the cases of flow past a sphere and self-propelled swimmers. We then presented the first-ever simulation of multiple 3D self-propelled swimmers, simulated on only four compute nodes, to highlight the flexibility and computing efficiency of the software.

13.2 Perspectives

The simulation techniques and results presented in this thesis provide several opportunities for future research. Here we will discuss some of the feasible extensions of our work, as well as possible new research directions.

Eulerian vs Lagrangian methods

In part I we described the rVM as a viable alternative to pseudo-spectral methods for the simulation of high Reynolds number incompressible flows. In fact, implementing the rVM with a fourth-order Runge-Kutta time stepping scheme and the M_6^* interpolation kernel achieves a similar accuracy as the spectral methods. One major difference between the two methods, and indeed the main source of discrepancy in their performance, is the time-stepping criterion. Pseudo-spectral methods are bound by an Eulerian CFL criterion, which is characterized by the maximum velocity in the domain as well as the minimum grid spacing. Instead the rVM is bound by a Lagrangian CFL criterion, which measures the deformation rate in the flow and is independent of the grid resolution.

In the simulations of chapter 4 we noticed that during reconnection events, characterized by large spatial gradients but not necessarily strong flow speeds, the LCFL criterion could be up to six times more restrictive than the CFL criterion. During quiescent, transport-dominated periods the LCFL criterion allowed up to four times larger time steps.

This implies that particle methods might not be well-suited for applications where the strain rate could consistently dominate over the transport terms, such as forced homogeneous isotropic turbulence. High Reynolds number FSI problems might have similar characteristics. There is scope for a closer investigation of the asymptotic difference between these two criterions as the Reynolds number increases, both for free-space vortical flows and for flows with boundary layers. This investigation needs to take into account the asymptotic trends for grid spacing requirements, maximum flow velocity and maximum strain rates.

If the strain rates in the domain are spatially localized, a possible solution to this issue could be to integrate particle-based advection with local time stepping (LTS) techniques as presented in section 9.2. Instead of grouping grid points by resolution they could be grouped by strain rate, and advected according to their own time scales. For higher order time integration techniques, where the velocity needs to be recomputed for every LTS sub-step, this requires a solver that can exploit the localization of these strain rate groups. A candidate is the multipole-based solver optimized for regular data structures as presented in chapter 10.

FSI algorithm

Parts II and III dealt with fluid-solid interaction problems, where we found that our algorithm can produce accurate results at the cost of small time steps. This defeats one of main advantages of particle methods, its stability at large time steps independent of the grid spacing. The cause of the limited temporal accuracy is twofold. Firstly, the algorithm presented in section 5.4 relies on first-order time splitting in order to compute the penalization term implicitly. Any explicit method would pose the problem of solving a right-hand side whose stiffness is dominated by λ, the penalization parameter. Secondly, the two-way coupling between fluid and solid is performed in a weak sense, so that they are updated in an alternating way rather than as a coupled system.

To alleviate only the first of these issues one could implement an iterative technique for the penalization term similar to Hejlesen, Koumoutsakos, et al. (2015). This is essentially another form of local time stepping, where only the support of the obstacle is considered.

The use of high-order explicit time-integration techniques for the combined right-hand side should not be discarded. This would facilitate the coupled advection of the fluid and solid, which increases the temporal order of accuracy for the entire FSI system. At first sight the extra velocity solver calls and the $\mathcal{O}(\lambda^{-1})$ time-stepping criterion might be inhibitive to such an approach. Yet it remains to be investigated whether the increased temporal accuracy could reduce the value of λ required in practice, and whether a larger time step could mitigate the cost of the additional velocity solver calls. Local time stepping together with multipole-type local velocity evaluation algorithms could also alleviate the stiffness in the equations. The resulting increased accuracy will in any case result in a larger memory footprint of the algorithm, since solutions at intermediate times will need to be stored.

We note that for FSI simulations using immersed techniques as presented in this thesis, the effect of a non-zero divergence in the vorticity field is unexplored. Our results were tested both with and without the divergence correction technique presented in section 2.3, and we did not observe significant effects. The small time steps required for accurate FSI simulations might also reduce the non-solenoidal error build-up in the vorticity field. Yet this might change when simulating higher Reynolds number flows and requires further investigation.

Structural mechanics

The solid bodies considered in this work do not contain any structural mechanics, instead their deformation is imposed at all times. In many practical applications, however, full fluid-structure interaction is desired, for instance when simulating the thin and flexible fins of fish. A typical and established approach to incorporate such effects are to combine an Eulerian fluid-solver with a Lagrangian, Finite-Element Method (FEM) type of solver. Such a coupling creates a difficulty, however, because the FEM explicitly discretizes a body's surface whereas typically modern fluid solvers work with implicit immersed surfaces.

An alternative, which has been explored in Sugiyama et al. (2011), is to cast the structural mechanics in the same Eulerian framework as the fluid mechanics. The strategy is to treat the system with a single-continuum approach to alleviate implementation and parallelization issues. In the simplest case the entire continuum is governed by incompressible dynamics and has unit density, so that one could consider the system

$$\nabla \cdot \mathbf{u} = 0 \tag{13.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot \big((1 - \chi_s)\boldsymbol{\sigma}_f + \chi_s \boldsymbol{\sigma}_s\big) \tag{13.2}$$

$$\frac{\partial \chi_s}{\partial t} + (\mathbf{u} \cdot \nabla)\chi_s = 0. \tag{13.3}$$

where χ_s is a characteristic function representing the solid domain and $\boldsymbol{\sigma}_f$ and $\boldsymbol{\sigma}_s$ denote the fluid and solid stress tensors. The solution of this system will require an implicit solution technique, due to the stiffness originating from the stress tensors terms, which needs further investigation. Furthermore, depending on the constitutive model for the solid, the evolution equation for the deformation gradient tensor $\mathbf{F}$ might be required, which reads

$$\frac{\partial \mathbf{F}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{F} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{F}. \tag{13.4}$$

This approach could be further extended by combining it with the penalization method of chapter 5 to represent fluid, deforming solid, and rigid solid materials and distinguish between, for instance, bones, tissue, and fins of a swimmer.

Biolocomotion

In Gazzola, Hejazialhosseini, and Koumoutsakos (2014) simulations of 2D self-propelled swimmers were coupled with a reinforcement learning framework (Sutton and Barto 1998). This enabled the study of optimal behavioral characteristics on top of the swimmer's morphology and kinematics studied here. The extension of this framework to 3D requires further work on each of the two sides of this coupling. For the reinforcement learning, the learning time of $\mathcal{O}(10^4)$ swimming cycles mentioned in Gazzola, Hejazialhosseini, and Koumoutsakos (2014) needs to be reduced through more effective discretizations of the learning matrix. Additionally, the time to solution could be reduced by implementing the learning process in a parallel framework of hydrodynamically independent simulations that simultaneously interact with the same policy. For 3D rVM-FSI solver, the time to solution needs to be reduced, which is discussed later in this chapter.

We have identified some of the vortex dynamics involved in 3D side-by-side swimming, although analysis of the detailed nature of the fluid-mediated interactions requires follow-up work. Yet the flexibility of the solver enables studying arbitrarily many swimmers in any 3D layout, with the only limit the computational resources. Such capabilities can provide unique possibilities to study the hydrodynamic benefits of fish schooling in a viscous fluid.

An extension of our current multiresolution rVM-FSI implementation to periodic boundary conditions would facilitate the simulations of infinite schools of swimmers without boundary effects, similar to the analytic low-order study of Weihs (1973a). The main work lies in extending our multipole-based velocity solver to periodic boundary conditions, for which the algorithm is readily available (L. Greengard and Rokhlin 1987), and to generalize the swimmer's implementation to account for periodic boundary crossings.

Finally, the counter-rotating cylinder pair studied in chapter 11 offers a simple propulsion technique. This work can be extended by modeling the behavior of the cylinders mathematically, to derive scaling laws and generalize the trends. A further extension is to study oscillatory rotations of the counter-rotating cylinders to investigate the relevant time scales required for such unsteady propulsion. In exploratory simulations we have noticed that this oscillatory model also reproduces in-line swimming by applying a phase shift between the two cylinders.

Software

The multiresolution 3D rVM software presented in part III provides a good basis for future work. Its accuracy, flexibility and performance has been shown in this thesis, and it was noted that the shared-memory velocity solver would further benefit from AVX vectorization for the indirect interactions and sharing the GPU between direct and indirect interactions.

The simulations in chapter 12, however, tend to consume current shared-memory resources entirely. At late simulation times the memory footprint can exceed the hardware resources, and the time-to-solution is too long for production simulations. Here we detail two non-exclusive solutions for this problem.

The first is to improve the efficiency of the representation of the flow. With the current MRAG design, simulations are restricted to domains of a unit cube. The wavelet-adaptivity overcomes this somewhat by coarsening away from the relevant region. We have seen in chapter 12 that dynamically adapting a uniform grid to the vorticity support can result in a similar number of computational elements as the wavelet adaptation in the unit cube. Yet the adaptive approach requires a larger memory footprint, as all the blocks require storage of their encoded ghost reconstruction patterns as detailed in section 9.3. It is therefore recommended to adapt the domain size to the vorticity support also for the adaptive-grid solver. The velocity solver and wavelets on the interval discussed in part III already offer the flexibility to handle non-rectangular domains, including the possibility of several disconnected groups of grid blocks scattered around the infinite real space. This, combined with the multiresolution approach, would result in the most compact representation possible.

The second option is to create a more advanced distributed-memory approach. This does not necessarily entail scalability up to massively parallel compute clusters, since the efficient usage of both memory and compute power offered by $\mathcal{O}(10)$ compute nodes would suffice for most practical problems. Specifically, this involves dynamically distributing the grid over the compute nodes, requiring a fast dynamic load balancer and an effective way of communicating data between compute nodes.

Part V

Appendix

APPENDIX A

Gallery

On the following pages we show seven stills that visualize various flow simulations performed during this thesis. All six figures were created using the VolumePerception software (Rossinelli 2011). Captions briefly explain the flow case and provide further references were appropriate.

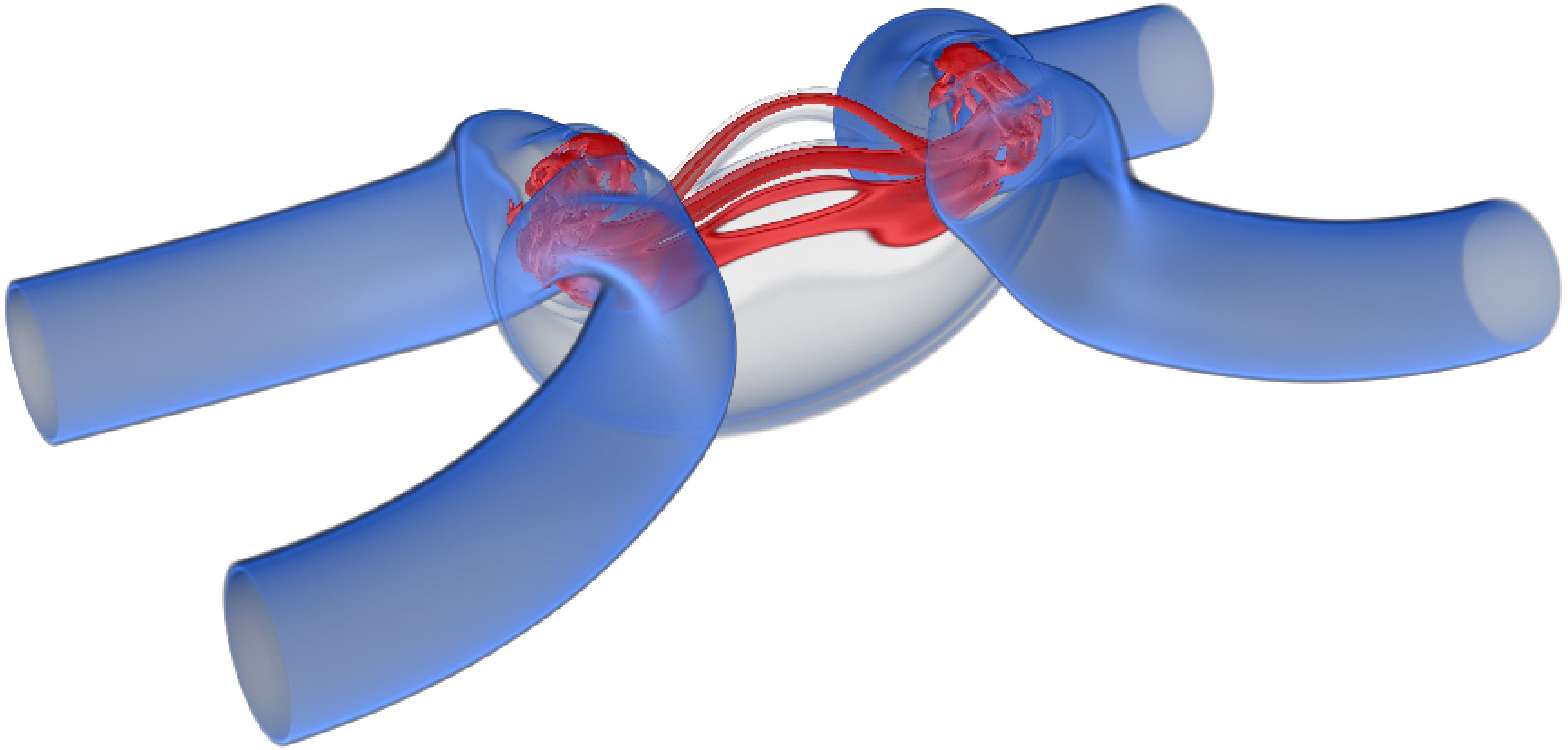


Figure A.1: Volume rendering of the vorticity field of two reconnecting vortex tubes without initial axial flow (case 0 of chapter 4, just after the first reconnection). Red/blue denoted high/low vorticity magnitude.

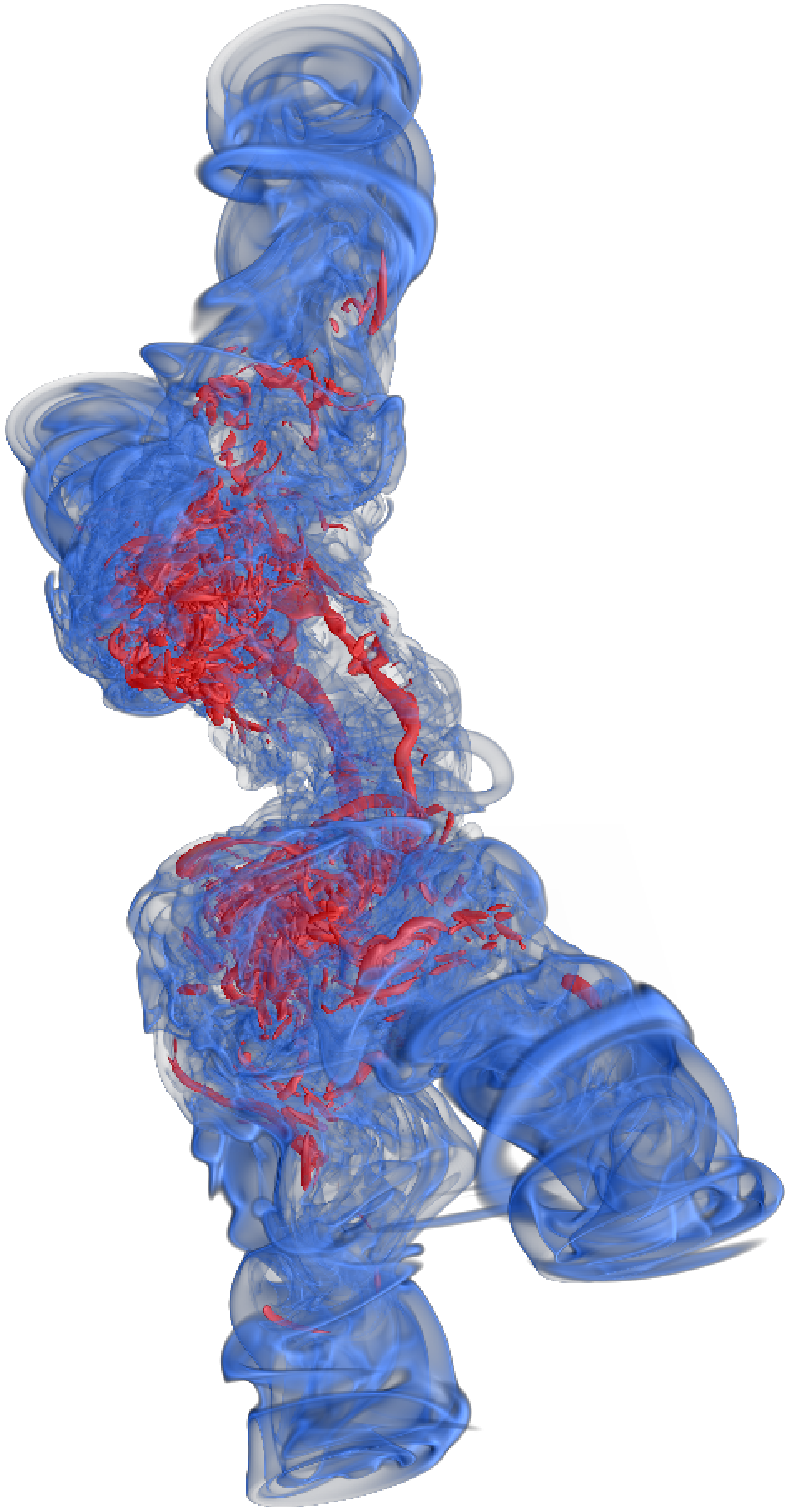


Figure A.2: Volume rendering of the vorticity field of two reconnecting vortex tubes without initial axial flow (case 0 of chapter 4, just after the second reconnection). Red/blue denoted high/low vorticity magnitude.

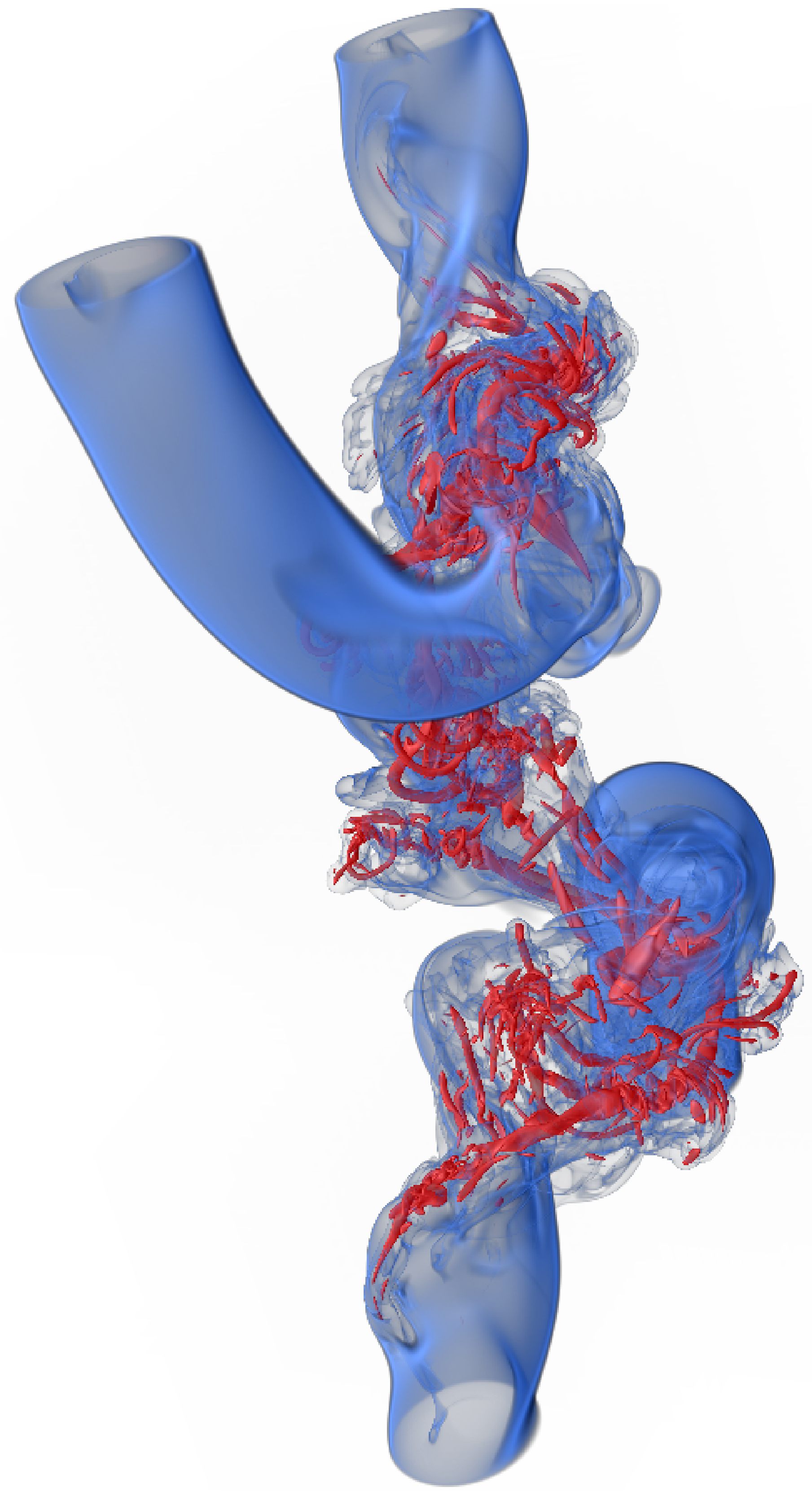


Figure A.3: Volume rendering of the vorticity field of two reconnecting vortex tubes with initial axial flow (case 2 of chapter 4, just after the first reconnection). Red/blue denoted high/low vorticity magnitude.

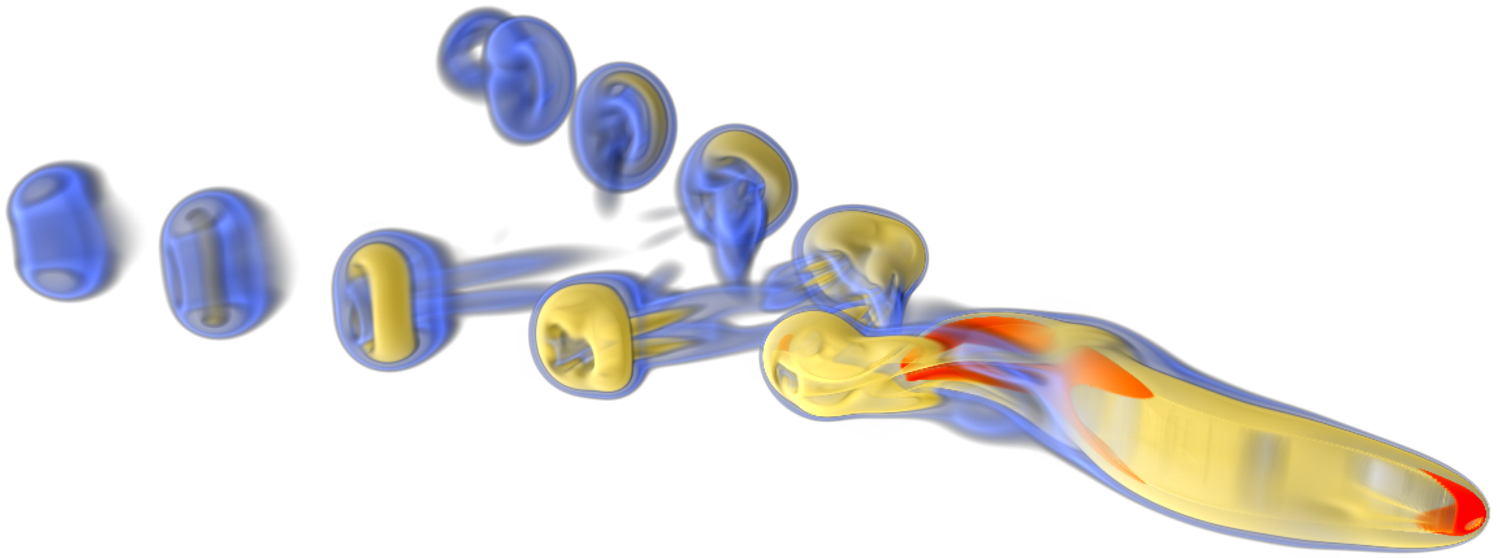


Figure A.4: Volume rendering of a self-propelled swimmer at $Re = 7143$. Red/yellow/blue denoted high/intermediate/low vorticity magnitude.

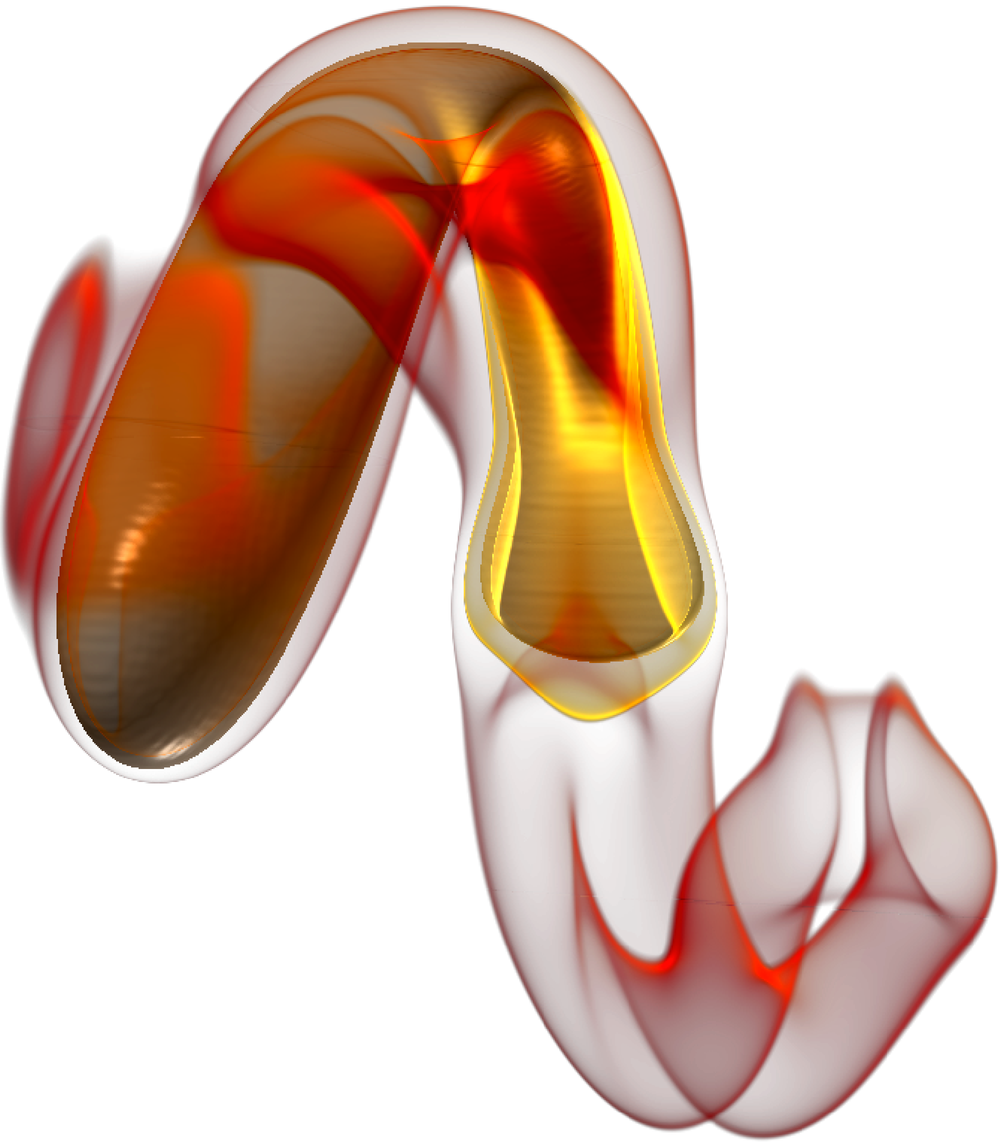


Figure A.5: Volume rendering of the vorticity field of a swimmer performing a C-start (chapter 6). Yellow/red denoted high/low vorticity magnitude.

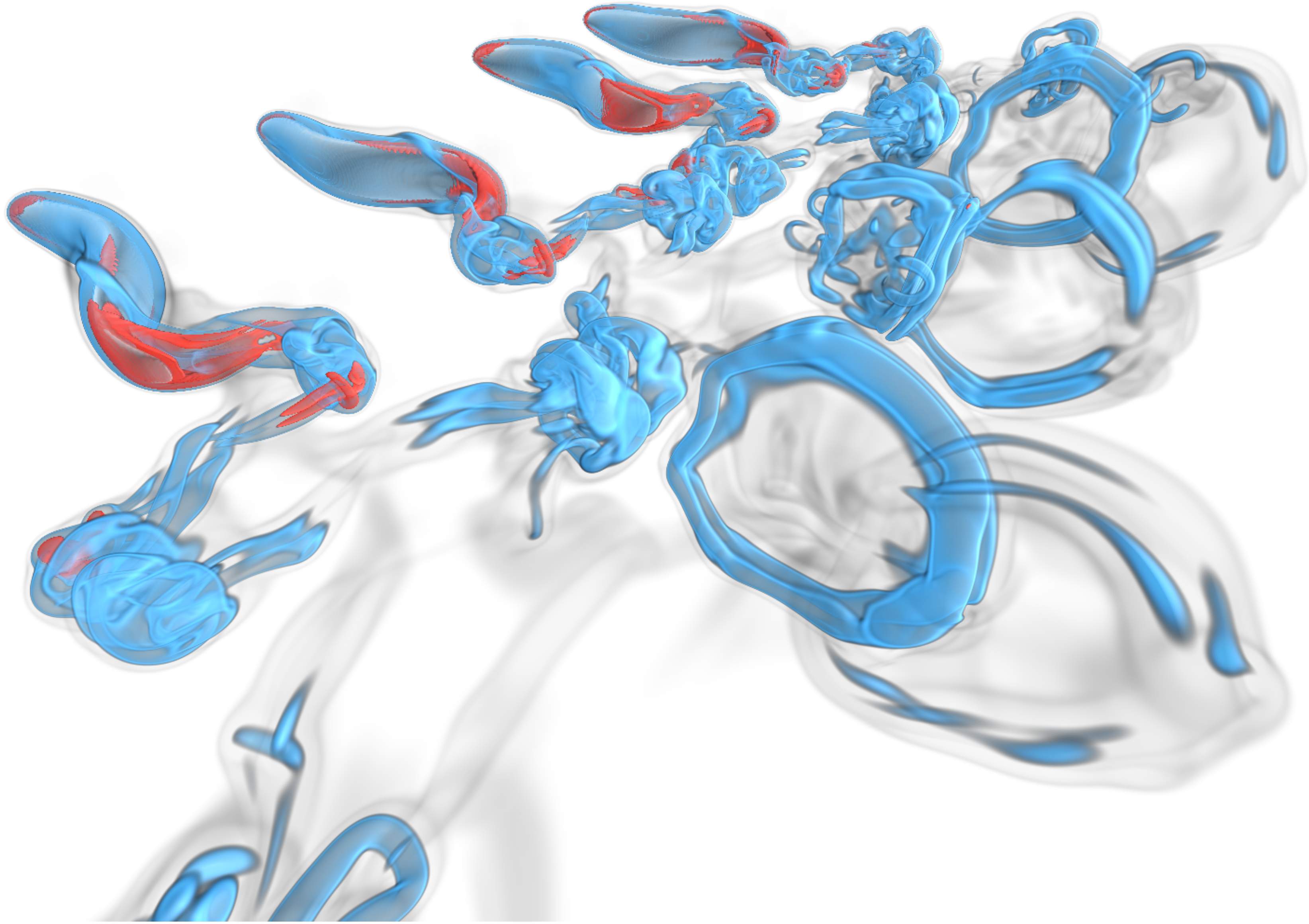


Figure A.6: Volume rendering of the vorticity field of a four side-by-side swimmers (chapter 12). Red/blue denoted high/low vorticity magnitude.

APPENDIX B

Numerical reconstruction of undulatory swimmers

All fluid-solid interaction simulations presented here involve the computation of the characteristic function χ and the deformation velocity field $\mathbf{u}_{\text{def}}$ on the Cartesian grid. For χ we have to discretize a Heaviside function onto the computational grid, which requires knowledge of the signed-distance function inside a narrow band of a small number of grid points around the shape boundary. For simple analytic geometries such as spheres and ellipsoids the function χ can be computed directly from the analytically known signed distance function. Similarly simple analytic deformation velocity fields such as imposed rigid-body translations and rotations can be computed directly on the grid.

For complicated geometries and deformations, such as undulatory swimmers, the procedure is more involved. Here we will describe how in our case the shapes and velocities are defined, and how we obtain the χ and $\mathbf{u}_{\text{def}}$ fields on the computational grid.

Our swimmers' geometries and deformation fields are based on an imposed planar midline deformation. The midline is parametrized with its arclength parameter s so that $0 \leq s \leq L$, where L is the swimmer's length. The midline coordinates at time t are given by the vector $\mathbf{x}_m(s,t) = (x_m(s,t), y_m(s,t))^T$, and the normal vectors pointing away from the midline are given as $\mathbf{n}_m(s,t) = (n_m^x(s,t), n_m^y(s,t))^T$. The velocities are then defined by $\mathbf{u}_m(s,t) = \dot{\mathbf{x}}_m(s,t)$.

In the rest of this chapter we will first consider the reconstruction of the midline deformation and deformation velocities. We then discuss the reconstruction of the entire 3D shape. The first step in the shape reconstruction is to obtain the signed-distance function and the deformation velocity field on a curvilinear grid. Afterwards we explain how to interpolate these characteristics between the curvilinear grid around the swimmer's midline and the regular cartesian computational grid.

B.1 Midline reconstruction

In this thesis we consider two different descriptions of the midline, which will be detailed in the next two sections. The first is based on an explicit definition of $y_m(s,t)$, whereas the second is based on a definition of the midline curvature $\kappa_m(s,t)$. In the latter case we need to take the additional step of solving the Frenet-Serret equations for a planar curve in order to obtain $\mathbf{x}_m(s,t)$ and $\mathbf{n}_m(s,t)$. In both cases we end up with the same information, that is the full (numerical) description of the midline positions, velocities and normals.

B.1.1 Explicit midline definition

The simplest definition of the midline kinematics involves an explicit function for the lateral position of the midline

$$y_m(s,t) = A(s)\sin\left(2\pi\left(\frac{t}{T} - \frac{s\tau}{L} + \phi\right)\right), \tag{B.1}$$

where $A(s)$ is the amplitude envelope, T is the swimming period, τ is the wavelength and ϕ is the phase shift of the traveling wave. Using the definition of the arclength parameter s we can obtain the corresponding $x_m(s,t)$ coordinates by integrating

$$x_m(s,t) = \int_0^s \sqrt{1 - \left(\frac{\partial y(\sigma,t)}{\partial\sigma}\right)}\,\mathrm{d}\sigma, \tag{B.2}$$

which can be solved using numerical quadrature as an initial value problem with starting condition $x_m(0,t) = 0$. The midline normal vector can then be found from

$$\mathbf{n}_m(s,t) = \begin{pmatrix} -y_m{}'(s,t) \\ x_m{}'(s,t) \end{pmatrix}, \tag{B.3}$$

where the prime denotes differentiation with respect to s.

B.1.2 Midline curvature definition

A different approach is to describe the curvature of the midline as a function of the arclength parameter

$$\kappa_m(s,t) = K(s)\sin\left(2\pi\left(\frac{t}{T} - \frac{s\tau}{L}\right)\right), \tag{B.4}$$

where $K(s)$ is the curvature distribution along the midline. We can then integrate the planar Frenet equations for the unit normal vectors $\mathbf{n}_m(s,t)$ and the unit tangential vectors

$\mathbf{t}_m(s,t)$, where $\mathbf{t}_m \times \mathbf{n}_m = \mathbf{1}$, as an initial value problem

$$\frac{\partial \mathbf{t}_m}{\partial s}(s,t) = \kappa_m(s,t)\mathbf{n}_m(s,t), \tag{B.5}$$

$$\frac{\partial \mathbf{n}_m}{\partial s}(s,t) = -\kappa_m(s,t)\mathbf{t}_m(s,t), \tag{B.6}$$

with initial conditions $\mathbf{n}_m(0,t) = (0,1)^T$ and $\mathbf{t}_m(0,t) = (1,0)^T$. By solving a second initial value problem we find the midline coordinates as a function of the tangent vectors

$$\frac{\partial \mathbf{x}_m}{\partial s}(s,t) = \mathbf{t}_m(s,t), \tag{B.7}$$

with initial conditions $x_m(0,t) = y_m(0,t) = 0$.

B.2 Shape reconstruction

For brevity, in the following we omit the temporal function argument t and assume we are working at one particular time instance. Given the midline coordinates $\mathbf{x}_m(s)$ and normals $\mathbf{n}_m(s)$ we can create a curvilinear grid with planar coordinates s along the midline and r normal to the midline. Since these coordinates are planar, let z be the vertical coordinate emanating from this plane. Then we can introduce θ as the angle between the z and the r axes, to obtain the curvilinear coordinates

$$\begin{aligned} x(s,r,\theta) &= x_m(s) + n_m^x(s) r\cos\theta, \\ y(s,r,\theta) &= y_m(s) + n_m^y(s) r\cos\theta, \\ z(s,r,\theta) &= r\sin\theta. \end{aligned} \tag{B.8}$$

For the velocities, we take the time derivative of the positions and normals to find

$$\begin{aligned} u(s,r,\theta) &= \dot{x}_m(s) + \dot{n}_m^x(s) r\cos\theta, \\ v(s,r,\theta) &= \dot{y}_m(s) + \dot{n}_m^y(s) r\cos\theta, \\ w(s,r,\theta) &= 0. \end{aligned} \tag{B.9}$$

The three-dimensional shape is then described with a width curve $w(s)$ and a height curve $h(s)$, that define the semi-axes of an elliptical cross-section. The volume is bounded by

$$0 \le s \le L, \tag{B.10}$$

$$0 \le r \le \frac{w(s)h(s)}{\sqrt{(h(s)\cos\theta)^2 + (w(s)\sin\theta)^2}} \equiv f(\theta,s), \tag{B.11}$$

$$0 \le \theta \le 2\pi, \tag{B.12}$$

where $f(\theta,s)$ corresponds to the radius of an ellipse with semi-axes $w(s)$ and $h(s)$ at angle θ.

B.2.1 Computing internal characteristics

To correctly position the fish onto the computational grid we first need translate the shape (essentially $\mathbf{x}_m(s,t)$) into the center of mass frame-of-reference. This requires finding the center of mass. Similarly, we need to compute the linear and angular momentum of the corresponding deformation velocity field to ensure that the $\mathbf{u}_{\text{def}}$ is momentum-free.

A generic approach for performing these steps has been described in Gazzola, Chatelain, et al. (2011). This approach relies on constructing numerically a curvilinear grid according to (B.8) on which the χ and $\mathbf{u}_{\text{def}}$ can readily be computed. Then it uses bilinear (in 2D) or trilinear (in 3D) interpolation to interpolate these onto a small regular Cartesian grid, separate from the main computational grid. Using simple midpoint quadrature rules the internal characteristics of the shape can then be integrated. These quantities are used to shift and rotate the shape and deformation velocity field according to the center of mass frame of reference, and to correctly account for its linear and angular momentum.

The memory overhead of these supporting grids on which the shape is evaluated is significant, especially for 3D computations with several swimmers. Here we describe a different strategy, based on semi-analytic integration of the relevant quantities in the curvilinear frame. This approach removes the requirement for the intermediate 3D deformation grid and reduces the computational footprint of the solver. Instead the relevant quantities are computed from several one-dimensional arrays that depend on s only. Here we show the specialization of the relevant integrals for elliptical cross-sections, though the equations could similarly be evaluated *a priori* for different geometries.

First, since $n_m^x(s)$ and $n_m^y(s)$ are the Cartesian components of a unit vector normal to the midline at location $x_m(s)$ and $y_m(s)$, the following identities apply

$$x_m{}'(s)^2 + y_m{}'(s)^2 = x_m{}'(s) n_m^y(s) - y_m{}'(s) n_m^x(s) = 1, \tag{B.13}$$

$$x_m{}'(s) n_m^y{}'(s) - y_m{}'(s) n_m^x{}'(s) = 0. \tag{B.14}$$

Furthermore, the Jacobian of the transformation between the Cartesian and the internal fish coordinates can be computed:

$$J = \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \\ \frac{\partial z}{\partial s} & \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} \end{pmatrix} = \begin{pmatrix} x_m{}'(s) + n_m^x{}'(s) r\cos\theta & n_m^x(s)\cos\theta & -n_m^x(s) r\sin\theta \\ y_m{}'(s) + n_m^y{}'(s) r\cos\theta & n_m^y(s)\cos\theta & -n_m^y(s) r\sin\theta \\ 0 & \sin\theta & r\cos\theta \end{pmatrix}, \tag{B.15}$$

with determinant

$$
\begin{aligned}
|J| &= (x_m{}'(s) + n_m^x{}'(s) r\cos\theta) n_m^y(s) r - (y_m{}'(s) + n_m^y{}'(s) r\cos\theta) n_m^x(s) r && \text{(B.16)}\\
&= r(x_m{}'(s) n_m^y(s) - y_m{}'(s) n_m^x(s)) + r^2\cos\theta (n_m^x{}'(s) n_m^y(s) - n_m^y{}'(s) n_m^x(s)) && \text{(B.17)}\\
&= r(n_m^y(s)^2 + n_m^x(s)^2) + r^2\cos\theta (n_m^x{}'(s) n_m^y(s) - n_m^y{}'(s) n_m^x(s)) && \text{(B.18)}\\
&= r + r^2\cos\theta (n_m^x{}'(s) n_m^y(s) - n_m^y{}'(s) n_m^x(s)) && \text{(B.19)}\\
&= r + r^2\cos\theta (x_m{}'(s) n_m^x{}'(s) + y_m{}'(s) n_m^y{}'(s)). && \text{(B.20)}
\end{aligned}
$$

We precomputed some results that will be used in the following, where $f(\theta, s)$ is defined is equation (B.11),

$$\int_0^{2\pi} f(\theta,s)^2 \, \mathrm{d}\theta = 2\pi w(s) h(s), \tag{B.21}$$

$$\int_0^{2\pi} f(\theta,s)^3 \cos(\theta) \, \mathrm{d}\theta = \int_0^{2\pi} f(\theta,s)^3 \sin(\theta) \, \mathrm{d}\theta = 0, \tag{B.22}$$

$$\int_0^{2\pi} f(\theta,s)^4 \cos^2(\theta) \, \mathrm{d}\theta = \pi w(s)^3 h(s), \tag{B.23}$$

$$\int_0^{2\pi} f(\theta,s)^4 \sin(\theta)\cos(\theta) \, \mathrm{d}\theta = 0, \tag{B.24}$$

$$\int_0^{2\pi} f(\theta,s)^5 \cos^3(\theta) \, \mathrm{d}\theta = 0, \tag{B.25}$$

and with these identities we can evaluate several useful 'moments' of the double integral of the Jacobian over r and θ:

$$\int_{\theta=0}^{2\pi} \int_{r=0}^{f(\theta,s)} |J| \, \mathrm{d}r \, \mathrm{d}\theta = \pi w(s) h(s), \tag{B.26}$$

$$\int_{\theta=0}^{2\pi} \int_{r=0}^{f(\theta,s)} r\sin\theta |J| \, \mathrm{d}r \, \mathrm{d}\theta = 0, \tag{B.27}$$

$$\int_{\theta=0}^{2\pi} \int_{r=0}^{f(\theta,s)} r\cos\theta |J| \, \mathrm{d}r \, \mathrm{d}\theta = \frac{\pi}{4} w(s)^3 h(s) \left(x_m{}'(s) n_m^x{}'(s) + y_m{}'(s) n_m^y{}'(s) \right), \tag{B.28}$$

$$\int_{\theta=0}^{2\pi} \int_{r=0}^{f(\theta,s)} r^2\cos^2\theta |J| \, \mathrm{d}r \, \mathrm{d}\theta = \frac{\pi}{4} w(s)^3 h(s). \tag{B.29}$$

With all these definitions and identities we can rewrite the volume, center of mass, moment of inertia and linear and angular momenta as integrals over s alone. These expressions can be readily integrated using simple numerical quadratures with very little overhead, both computationally and in terms of memory footprint. The full expressions are given in the following.

Volume

$$V = \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} |J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s = \pi\int_{s=0}^{L} w(s)h(s)\,\mathrm{d}s.$$

Center of mass

$$\begin{aligned}
V\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix} &= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \begin{pmatrix}x(s,r,\theta)\\ y(s,r,\theta)\end{pmatrix}|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \begin{pmatrix}x_m(s)+n_m^x(s)r\cos\theta\\ y_m(s)+n_m^y(s)r\cos\theta\end{pmatrix}|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \pi\int_{s=0}^{L}\begin{pmatrix}x_m(s)\\ y_m(s)\end{pmatrix} w(s)h(s)\,\mathrm{d}s + \frac{\pi}{4}\int_{s=0}^{L}\begin{pmatrix}n_m^x(s)\\ n_m^y(s)\end{pmatrix}\left(n_m^x{}'(s)n_m^y(s) - n_m^x(s)n_m^y{}'(s)\right)w(s)^3h(s)\,\mathrm{d}s.
\end{aligned}$$

Moment of inertia

$$\begin{aligned}
J &= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \left(x(s,r,\theta)^2 + y(s,r,\theta)^2\right)|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \left((x_m(s)+n_m^x(s)r\cos\theta)^2 + (y_m(s)+n_m^y(s)r\cos\theta)^2\right)|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \pi\int_{s=0}^{L}\left(x_m(s)^2 + y_m(s)^2\right)w(s)h(s)\,\mathrm{d}s\\
&+ \frac{\pi}{2}\int_{s=0}^{L}\left(x_m(s)n_m^x(s) + y_m(s)n_m^y(s)\right)w(s)^3h(s)\left(n_m^x{}'(s)n_m^y(s) - n_m^y{}'(s)n_m^x(s)\right)\,\mathrm{d}s\\
&+ \frac{\pi}{4}\int_{s=0}^{L} w(s)^3h(s)\,\mathrm{d}s.
\end{aligned}$$

Linear momentum

$$\begin{aligned}
\begin{pmatrix}L_x\\ L_y\end{pmatrix} &= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \begin{pmatrix}u(s,r,\theta)\\ v(s,r,\theta)\end{pmatrix}|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \begin{pmatrix}\dot{x}_m(s)+\dot{n}_m^x(s)r\cos\theta\\ \dot{y}_m(s)+\dot{n}_m^y(s)r\cos\theta\end{pmatrix}|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s\\
&= \pi\int_{s=0}^{L}\begin{pmatrix}\dot{x}_m(s)\\ \dot{y}_m(s)\end{pmatrix} w(s)h(s)\,\mathrm{d}s + \frac{\pi}{4}\int_{s=0}^{L}\begin{pmatrix}\dot{n}_m^x(s)\\ \dot{n}_m^y(s)\end{pmatrix}\left(n_m^x{}'(s)n_m^y(s) - n_m^x(s)n_m^y{}'(s)\right)w(s)^3h(s)\,\mathrm{d}s.
\end{aligned}$$

Angular momentum

$$
\begin{aligned}
I &= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} \left(x(s,r,\theta)v(s,r,\theta) - y(s,r,\theta)u(s,r,\theta)\right)|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s \\
&= \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} (x_m(s) + n_m^x(s) r\cos\theta)(\dot{y}_m(s) + \dot{n}_m^y(s) r\cos\theta)|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s \\
&- \int_{s=0}^{L}\int_{\theta=0}^{2\pi}\int_{r=0}^{f(\theta,s)} (y_m(s) + n_m^y(s) r\cos\theta)(\dot{x}_m(s) + \dot{n}_m^x(s) r\cos\theta)|J|\,\mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}s \\
&= \pi\int_{s=0}^{L} (x_m(s)\dot{y}_m(s) - y_m(s)\dot{x}_m(s))\,w(s)h(s)\,\mathrm{d}s \\
&+ \frac{\pi}{4}\int_{s=0}^{L} \left(x_m(s)\dot{n}_m^y(s) - y_m(s)\dot{n}_m^x(s)\right) w(s)^3 h(s) \left(n_m^{x\,\prime}(s) n_m^y(s) - n_m^{y\,\prime}(s) n_m^x(s)\right)\,\mathrm{d}s \\
&+ \frac{\pi}{4}\int_{s=0}^{L} \left(\dot{y}_m(s) n_m^x(s) - \dot{x}_m(s) n_m^y(s)\right) w(s)^3 h(s) \left(n_m^{x\,\prime}(s) n_m^y(s) - n_m^{y\,\prime}(s) n_m^x(s)\right)\,\mathrm{d}s \\
&+ \frac{\pi}{4}\int_{s=0}^{L} \left(n_m^x(s)\dot{n}_m^y(s) - n_m^y(s)\dot{n}_m^x(s)\right) w(s)^3 h(s)\,\mathrm{d}s.
\end{aligned}
$$

B.2.2 Interpolation onto computational domain

After the shape midlines and deformation velocity field are adjusted to the center of mass frame-of-reference, the characteristic function χ and the deformation velocity field $\mathbf{u}_{\mathrm{def}}$ have to be set on the computational domain. In this step the discrete representation of the swimmer has to be changed from curvilinear to cartesian grids. For this we use some of the interpolation tools in our particle-mesh framework. The deformation velocity field $\mathbf{u}_{\mathrm{def}}$ can be interpolated with any of the available particle-to-mesh (P2M) interpolation kernels as described in section 2.2.

For the characteristic function χ, we distinguish between the narrow-band along the interface, where χ smoothly varies between 0 and 1, and the region away from the interface, where χ is constant. For the latter, we use fast heuristic methods to determine whether a grid point is inside or outside of the body, and set χ accordingly. Near the interface, we require knowledge of the signed distance function (SDF) to mollify the characteristic function. The SDF is computed on the computational grid by applying a modified P2M operation to the point cloud that describes the three-dimensional surface (Diego Rossinelli, personal communication). The kernel in this operation simply evaluates the Euclidean distance within a narrow-band

$$D(r) = \begin{cases} r & \text{if } r < \mathcal{N}, \\ \infty & \text{else.} \end{cases} \tag{B.30}$$

where $\mathcal{N}$ is the thickness of the narrow band. In the P2M operation we now replace the traditional sum with a minimum operator, and obtain the Euclidean distance function (EDF) by evaluating

$$\mathrm{EDF}(\mathbf{x}) = \min_{p \in P} D(\left|\left|\mathbf{x} - \mathbf{x}_p\right|\right|_2), \tag{B.31}$$

where P denotes the set of particles that form the point cloud. This evaluation step can be transformed into a scatter-type operation by looping over the particles and considering only the grid points within the support dictated by $\mathcal{N}$. As a last step, one needs to apply the sign to the resulting function to obtain the full signed-distance function within the narrow band on the computational grid points. This approach works for both uniform and multiresolution computational grids.

APPENDIX C

Representing 3D rotations with quaternions

The fluid-solid interactions discussed in part II are limited to planar motions of the obstacle. For full 3D translations and rotations, a suitable implementation of 3D angular motions is required. In the multiresolution remeshed vortex code discussed in part III, we therefore implemented 3D rotations using quaternions. This chapter serves to give a short explanation and some details on how we use quaternions in our software.

Quaternions span the set of $\mathbb{R}^4$, extending the complex number system by using basis elements denoted as 1 (the real part) and i, j and k (the imaginary parts). A quaternion vector $\mathbf{q}$ can be therefore be written

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k. \tag{C.1}$$

The basis elements are related by the fundamental identities

$$i^2 = j^2 = k^2 = ijk = -1. \tag{C.2}$$

A 3D vector $\mathbf{u} = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$, where $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ are the Cartesian unit vectors, can be represented by a quaternion with a zero real part:

$$\mathbf{u}_q = u_1 i + u_2 j + u_3 k. \tag{C.3}$$

Like a standard complex number can describe a 2D rotation, a unit-length quaternion $\mathbf{q}$ can represent a 3D rotation. The quaternion elements describe a rotation angle θ around an axis $p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}$ as follows

$$\mathbf{q} = e^{\theta/2(p_1 i + p_2 j + p_3 k)} = \cos\left(\frac{\theta}{2}\right) + (p_1 i + p_2 j + p_3 k)\sin\left(\frac{\theta}{2}\right), \tag{C.4}$$

thereby enforcing $|\mathbf{q}| = 1$ to eliminate one degree of freedom. Using the algebra of quaternions this representation provides a way to, for instance, rotate existing vectors or concatenating series of rotations.

When computing the angular velocities of the object using our scheme, we obtain an angular velocity vector $\boldsymbol{\omega}_s$. Therefore the quaternion of the object has to be integrated in time according to this angular velocity. However for a quaternion to represent a valid rotation, it has to be remain a unit length quaternion, so the time integration has to respect this constraint. In our multiresolution code we have implemented a geometrically-corrected quaternion time integration as introduced in Kleppmann (2007).

Finally, when using quaternions to characterize the angular orientation of an object, one often needs its conversion to a 3D rotation matrix. This conversion is given as

$$\begin{pmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_2q_3+q_0q_1) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{pmatrix}. \tag{C.5}$$

APPENDIX D

Efficiency of self-propelled swimmers

Here we further detail the definition of efficiency used in this work, and how the relevant quantities are extracted from the numerical simulations. The efficiency definition used in this work is

$$f_{\text{eff}} = -\frac{E_{\text{useful}}}{E_{\text{input}} + E_{\text{useful}}} = -\frac{m\overline{U}^2/2}{\left(\int_{5T}^{6T} P_{\text{input}}(t)\,\mathrm{d}t\right) + m\overline{U}^2/2}, \tag{D.1}$$

where m is the mass of the swimmer and $\overline{U}$ is the average forward velocity, defined as

$$\overline{U} = \left\| \frac{1}{T} \int_{5T}^{6T} \mathbf{U}(t)\,\mathrm{d}t \right\|_2 , \tag{D.2}$$

where $\mathbf{U}$ is the velocity vector of the swimmer's center of mass. P_{input} is the total instantaneous power delivered to the fluid, which accounts for the rate of change of kinetic energy and dissipation due to viscous stresses. In the following we wish to clarify the definition and computation of P_{input}.

D.1 Definition of input power

We consider the total instantaneous power delivered to the flow as

$$P_{\text{input}} = \int_{\partial\Omega_s} (\mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{u})\,\mathrm{d}S. \tag{D.3}$$

Applying Gauss' theorem to this integral gives

$$P_{\text{input}} = \int_{\Omega_f} (\nabla \cdot \boldsymbol{\sigma} \cdot \mathbf{u})\,\mathrm{d}V \tag{D.4}$$

$$= \int_{\Omega_f} (\mathbf{u}(\nabla \cdot \boldsymbol{\sigma}) + \boldsymbol{\sigma} : \nabla \mathbf{u})\,\mathrm{d}V. \tag{D.5}$$

The first term of the second equation can be obtained by dotting the Navier-Stokes equation with the velocity vector, whereas for the second term we can substitute the definition of the stress tensor $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$

$$P_{\text{input}} = \int_{\Omega_f} \left(\rho \frac{\mathrm{D}}{\mathrm{D}t}\frac{u^2}{2} - p\nabla \cdot \mathbf{u} + \boldsymbol{\tau} : \nabla \mathbf{u}\right) \mathrm{d}V, \tag{D.6}$$

where $u^2 = \mathbf{u} \cdot \mathbf{u}$. Using the incompressibility condition of our fluid, $\nabla \cdot \mathbf{u} = 0$, and the Newtonian shear stress tensor $\boldsymbol{\tau} = \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ with μ the kinematic viscosity, we finally get

$$P_{\text{input}} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_f} \rho \frac{u^2}{2}\,\mathrm{d}V + \mu \int_{\Omega_f} \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) : \nabla\mathbf{u}\,\mathrm{d}V. \tag{D.7}$$

This is the formulation used in our work to compute the input power.

D.2 Computation of input power

As detailed in the main text of this work, our numerical method is based on the vorticity-velocity formulation of the Navier-Stokes equations with free-space boundary conditions. This means that the velocity field, which physically spans the entire free-space, is not entirely contained within the computational domain. Consequently, the evaluation of the input power P_{input} (equation D.7) needs to be carefully approached.

Concerning the first term in equation (D.7) we note that, for a divergence-free velocity field, the following kinematic identity holds

$$\int_{\Omega_f} \mathbf{u} \cdot \mathbf{u}\,\mathrm{d}V = \int_{\Omega_f} \boldsymbol{\Psi} \cdot \boldsymbol{\omega}\ \mathrm{d}V. \tag{D.8}$$

Here $\boldsymbol{\Psi}$ is the streamfunction, defined as the solution of the Poisson equation

$$\Delta\boldsymbol{\Psi} = -\boldsymbol{\omega}, \tag{D.9}$$

hence $\mathbf{u} = \nabla \times \boldsymbol{\Psi}$. The integral on the right-hand side of equation (D.8) can be computed in Fourier space (Chatelain and Koumoutsakos 2010) from a compact vorticity field, and thus the kinetic energy in a domain with free-space boundary conditions can be computed given only the vorticity field. In the current case, due to the deformation velocity field of the swimmer, the velocity field inside the body is not divergence free, and therefore this approach needs to be amended. The velocity field can be expressed via the Helmholtz-Hodge decomposition

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi} + \nabla \phi, \tag{D.10}$$

with

$$\Delta \phi = \nabla \cdot \mathbf{u}, \tag{D.11}$$

where the field $\nabla \cdot \mathbf{u}$ is non-zero only within the swimmer's support. The integral equation for the kinetic energy then expands into three contributions

$$\int_{\Omega_f} \mathbf{u} \cdot \mathbf{u} \, \mathrm{d}V = \int_{\Omega_f} \{(\nabla \times \boldsymbol{\Psi} \cdot \nabla \times \boldsymbol{\Psi}) + (\nabla \phi \cdot \nabla \phi) + 2(\nabla \times \boldsymbol{\Psi} \cdot \nabla \phi)\} \, \mathrm{d}V. \tag{D.12}$$

All three of these integrals can be computed in Fourier space based on compact fields in physical space ($\boldsymbol{\omega}$ and $\nabla \cdot \mathbf{u}$), and the sum of these three integrals results in the total kinetic energy in the domain. Since we need to compute the kinetic energy only in the fluid domain, we subtract the kinetic energy within the solid shape, easily calculated in physical space, from this sum. Finally, the derivative of the integral is computed with a first order finite difference technique applied between two subsequent time steps.

The second integral in equation D.7 represents the viscous dissipation term. We argue that due to the low Reynolds number and the corresponding strong decay of the vorticity away from the swimmer, the velocity gradients $\nabla \mathbf{u}$ outside our domain give negligible contributions to the integral, and can be ignored in the computation for the input power.

The convergence of this approach has been verified through successive domain increases and the assumptions were found to have a negligible influence on the value of the computed efficiency.

References

Alam, J. M., N. K.-R. Kevlahan, and O. V. Vasilyev (2006). "Simultaneous space-time adaptive wavelet solution of nonlinear parabolic differential equations." In: *Journal of Computational Physics* 214.2, pp. 829–857. DOI: 10.1016/j.jcp.2005.10.009.

Alben, S. and M. J. Shelley (2005). "Coherent locomotion as an attracting state for a free flapping body." In: *Proceedings of the National Academy of Sciences of the United States of America* 102.32, pp. 11163–11166. DOI: 10.1073/pnas.0505064102.

Almgren, A. S., J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome (1998). "A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations." In: *Journal of Computational Physics* 142.1, pp. 1–46. DOI: 10.1006/jcph.1998.5890.

Anderson, C. and C. Greengard (1985). "On vortex methods." In: *SIAM Journal on Numerical Analysis* 22.3, pp. 413–440. DOI: 10.1137/0722025.

Angot, P., C.-H. Bruneau, and P. Fabrie (1999). "A penalization method to take into account obstacles in incompressible viscous flows." In: *Numerische Mathematik* 81.4, pp. 497–520. DOI: 10.1007/s002110050401.

Aref, H. and I. Zawadzki (1991). "Linking of vortex rings." In: *Nature* 354.6348, pp. 50–53. DOI: 10.1038/354050a0.

Arquis, E. and J.-P. Caltagirone (1984). "On the hydrodynamical boundary-conditions along a fluid layer porous-medium interface - Application to the case of free convection." In: *Comptes Rendus de l Academie des Sciences Serie II* 299.1, pp. 1–4.

Ashurst, W. and D. I. Meiron (1987). "Numerical study of vortex reconnection." In: *Physical Review Letters* 58.16, pp. 1632–1635. DOI: 10.1103/PhysRevLett.58.1632.

Barnes, J. and P. Hut (1986). "A hierarchical O(N log N) force-calculation algorithm." In: *Nature* 324.6096, pp. 446–449. DOI: 10.1038/324446a0.

Bartol, I. K., M. Gharib, P. W. Webb, D. Weihs, and M. S. Gordon (2005). "Body-induced vortical flows: a common mechanism for self-corrective trimming control in boxfishes." In: *Journal of Experimental Biology* 208, pp. 327–344. DOI: 10.1242/jeb.01356.

Bergdorf, M. (2007). "Multiresolution Particle Methods for the Simulation of Growth and Flow." Doctoral dissertation. ETH, Zürich. DOI: 10.3929/ethz-a-005565040.

Bergdorf, M., G.-H. Cottet, and P. Koumoutsakos (2005). "Multilevel Adaptive Particle Methods for Convection-Diffusion Equations." In: *Multiscale Modeling & Simulation* 4.1, pp. 328–357. DOI: 10.1137/040602882.

Bergdorf, M. and P. Koumoutsakos (2006). "A Lagrangian Particle-Wavelet Method." In: *Multiscale Modeling & Simulation* 5.3, pp. 980–995. DOI: 10.1137/060652877.

Bergdorf, M., P. Koumoutsakos, and A. Leonard (2007). "Direct numerical simulations of vortex rings at $Re_\Gamma = 7500$." In: *Journal of Fluid Mechanics* 581, pp. 495–505. DOI: 10.1017/S0022112007006192.

Berger, M. J. and J. Oliger (1984). "Adaptive mesh refinement for hyperbolic partial differential equations." In: *Journal of Computational Physics* 53.3, pp. 484–512. DOI: 10.1016/0021-9991(84)90073-1.

Bhalla, A. P. S., R. Bale, B. E. Griffith, and N. A. Patankar (2013). "A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies." In: *Journal of Computational Physics* 250, pp. 446–476. DOI: 10.1016/j.jcp.2013.04.033.

Blake, R. W. (2004). "Fish functional design and swimming performance." In: *Journal of Fish Biology* 65.5, pp. 1193–1222. DOI: 10.1111/j.0022-1112.2004.00568.x.

Blumofe, R. D., C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, and Y. Zhou (1996). "Cilk: An Efficient Multithreaded Runtime System." In: *Journal of Parallel and Distributed Computing* 37.1, pp. 55–69. DOI: 10.1006/jpdc.1996.0107.

Blumofe, R. D. and C. E. Leiserson (1999). "Scheduling multithreaded computations by work stealing." In: *Journal of the ACM* 46.5, pp. 720–748. DOI: 10.1145/324133.324234.

Board, O. A. R. (2008). *OpenMP Application Program Interface*. Tech. rep. OpenMP Architecture Review Board.

Boratav, O., R. Pelz, and N. J. Zabusky (1992). “Reconnection in orthogonally interacting vortex tubes: Direct numerical simulations and quantifications.” In: *Physics of Fluids A: Fluid Dynamics* 4.3, pp. 581–605. DOI: 10.1063/1.858329.

Borazjani, I. and F. Sotiropoulos (2009). “Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes.” In: *Journal of Experimental Biology* 212.Pt 4, pp. 576–592. DOI: 10.1242/jeb.025007.

Borazjani, I. and F. Sotiropoulos (2010). “On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming.” In: *Journal of Experimental Biology* 213.1, pp. 89–107. DOI: 10.1242/jeb.030932.

Borazjani, I., F. Sotiropoulos, E. D. Tytell, and G. V. Lauder (2012). “Hydrodynamics of the bluegill sunfish C-start escape response: three-dimensional simulations and comparison with experimental data.” In: *Journal of Experimental Biology* 215.Pt 4, pp. 671–684. DOI: 10.1242/jeb.063016.

Borue, V. and S. A. Orszag (1997). “Spectra in helical three-dimensional homogeneous isotropic turbulence.” In: *Physical Review E* 55.6, pp. 7005–7009. DOI: 10.1103/PhysRevE.55.7005.

Boyd, J. P. (2000). *Chebyshev and Fourier Spectral Methods*. 2nd. Dover Publications.

Brachet, M. E., D. I. Meiron, S. A. Orszag, B. G. Nickel, R. H. Morf, and U. Frisch (1983). “Small-scale structure of the Taylor-Green vortex.” In: *Journal of Fluid Mechanics* 130, pp. 411–452. DOI: 10.1017/S0022112083001159.

Brissaud, A., U. Frisch, J. Leorat, M. Lesieur, and A. Mazure (1973). “Helicity cascades in fully developed isotropic turbulence.” In: *Physics of Fluids* 16.8, p. 1366. DOI: 10.1063/1.1694520.

Buche, D., P. Stoll, R. Dornberger, and P. Koumoutsakos (2002). “Multiobjective evolutionary algorithm for the optimization of noisy combustion processes.” In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 32.4, pp. 460–473. DOI: 10.1109/TSMCB.2002.804372.

Budick, S. A. and D. M. O’Malley (2000). “Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture.” In: *Journal of Experimental Biology* 203.17, pp. 2565–2579.

Cabral, B. and L. C. Leedom (1993). “Imaging vector fields using line integral convolution.” In: *Proceedings of the 20th annual conference on Computer graphics and interactive* *techniques - SIGGRAPH '93*. New York, New York, USA: ACM Press, pp. 263–270. ISBN: 0897916018. DOI: 10.1145/166117.166151.

Canuto, C., M. Y. Hussaini, A. Quarteroni, and T. A. Zang (2006). *Spectral Methods. Fundamentals in Single Domains*. Springer-Verlag, Berlin.

Canuto, C., M. Y. Hussaini, A. Quarteroni, and T. A. Zang (2007). *Spectral Methods. Evolution to complex geometries and applications to fluid dynamics*. Springer-Verlag, Berlin.

Carbou, G. and P. Fabrie (2003). "Boundary layer for a penalization method for viscous incompressible flow." In: *Advances in Difference Equations* 8, pp. 1453–1480.

Carling, J., T. L. Williams, and G. Bowtell (1998). "Self-propelled anguilliform swimming: Simultaneous solution of the two-dimensional Navier-Stokes equations and Newton's laws of motion." In: *Journal of Experimental Biology* 201.23, pp. 3143–3166.

Carnevale, G. F., M. Briscolin, R. C. Kloosterziel, and G. K. Vallis (1997). "Three-dimensionally perturbed vortex tubes in a rotating flow." In: *Journal of Fluid Mechanics* 341, pp. 127–163. DOI: 10.1017/S0022112097005430.

Chan, A. S., P. A. Dewey, A. Jameson, C. Liang, and A. J. Smits (2011). "Vortex suppression and drag reduction in the wake of counter-rotating cylinders." In: *Journal of Fluid Mechanics* 679, pp. 343–382. DOI: 10.1017/jfm.2011.134.

Chatelain, P., A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, and P. Koumoutsakos (2008). "Billion vortex particle direct numerical simulations of aircraft wakes." In: *Computer Methods in Applied Mechanics and Engineering* 197.13-16, pp. 1296–1304. DOI: 10.1016/j.cma.2007.11.016.

Chatelain, P., M. Gazzola, S. Kern, and P. Koumoutsakos (2011). "Optimization of Aircraft Wake Alleviation Schemes through an Evolution Strategy." In: *High Performance Computing for Computational Science – VECPAR 2010*. Lecture Notes in Computer Science 6449. Ed. by J. M. L. M. Palma, M. Daydé, O. Marques, and J. C. Lopes, pp. 210–221. DOI: 10.1007/978-3-642-19328-6.

Chatelain, P., D. Kivotides, and A. Leonard (2003). "Reconnection of Colliding Vortex Rings." In: *Physical Review Letters* 90.5, p. 054501. DOI: 10.1103/PhysRevLett.90.054501.

Chatelain, P. and P. Koumoutsakos (2010). "A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions." In: *Journal of Computational Physics* 229.7, pp. 2425–2431. DOI: 10.1016/j.jcp.2009.12.035.

Chen, Q., S. Chen, and G. L. Eyink (2003). “The joint cascade of energy and helicity in three-dimensional turbulence.” In: *Physics of Fluids* 15.2, p. 361. DOI: 10.1063/1.1533070.

Childress, S. (1981). *Mechanics of swimming and flying*. Cambridge University Press.

Chorin, A. J. (1973). “Numerical study of slightly viscous flow.” In: *Journal of Fluid Mechanics* 57.4, pp. 785–796. DOI: 10.1017/S0022112073002016.

Chu, C.-C., C.-T. Wang, C.-C. Chang, R.-Y. Chang, and W.-T. Chang (1995). “Head-on collision of two coaxial vortex rings: experiment and computation.” In: *Journal of Fluid Mechanics* 296, pp. 39–71. DOI: 10.1017/S0022112095002060.

Cocle, R., L. Bricteux, and G. Winckelmans (2009). “Scale dependence and asymptotic very high Reynolds number spectral behavior of multiscale subgrid models.” In: *Physics of Fluids* 21.8, p. 085101. DOI: 10.1063/1.3194302.

Cohen, A. (2002). “Adaptive methods for PDE’s: wavelets or mesh refinement?” In: *Proceedings of the International Congress of Mathematicians, Beijing*. Vol. 1. Higher Education Press, pp. 607–620.

Cohen, A., I. Daubechies, and J.-C. Feauveau (1992). “Biorthogonal bases of compactly supported wavelets.” In: *Communications on Pure and Applied Mathematics* 45.5, pp. 485–560. DOI: 10.1002/cpa.3160450502.

Conte, J., Y. Modarres-Sadeghi, M. N. Watts, F. S. Hover, and M. S. Triantafyllou (2010). “A fast-starting mechanical fish that accelerates at 40 m s(-2).” In: *Bioinspiration & biomimetics* 5.3, p. 035004. DOI: 10.1088/1748-3182/5/3/035004.

Contreras, G. and M. Martonosi (2008). “Characterizing and improving the performance of Intel Threading Building Blocks.” In: *2008 IEEE International Symposium on Workload Characterization*. Piscataway, NJ, USA: IEEE, pp. 57–66. ISBN: 978-1-4244-2777-2. DOI: 10.1109/IISWC.2008.4636091.

Coquerelle, M. and G.-H. Cottet (2008). “A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies.” In: *Journal of Computational Physics* 227.21, pp. 9121–9137. DOI: 10.1016/j.jcp.2008.03.041.

Corporation, N. (2012). *{CUDA} {C} Programming Guide v5.0*.

Cottet, G.-H., J.-M. Etancelin, F. Perignon, and C. Picard (2014). “High order semi-Lagrangian particle methods for transport equations: numerical analysis and implementation issues.” In: *ESAIM: Mathematical Modelling and Numerical Analysis* 48.4, pp. 1029–1060. DOI: 10.1051/m2an/2014009.

Cottet, G.-H. and P. Koumoutsakos (2000). *Vortex Methods, Theory and Practice*. Cambridge University Press.

Cottet, G.-H., B. Michaux, S. Ossia, and G. VanderLinden (2002). "A Comparison of Spectral and Vortex Methods in Three-Dimensional Incompressible Flows." In: *Journal of Computational Physics* 175.2, pp. 702–712. DOI: 10.1006/jcph.2001.6963.

Cottet, G.-H. and L. Weynans (2006). "Particle methods revisited: a class of high order finite-difference methods." In: *Comptes Rendus Mathematique* 343.1, pp. 51–56. DOI: 10.1016/j.crma.2006.05.001.

Crow, S. C. (1970). "Stability theory for a pair of trailing vortices." In: *AIAA Journal* 8.12, pp. 2172–2179. DOI: 10.2514/3.6083.

Cuypers, Y., A. Maurel, and P. Petitjeans (2003). "Vortex Burst as a Source of Turbulence." In: *Physical Review Letters* 91.19, p. 194502. DOI: 10.1103/PhysRevLett.91.194502.

Deslauriers, G. and S. Dubuc (1989). "Symmetric Iterative Interpolation Processes." In: *Constructive Approximation*. Ed. by R. A. DeVore and E. B. Saff. Boston, MA: Springer US, pp. 49–68. ISBN: 978-1-4899-6816-6. DOI: 10.1007/978-1-4899-6886-9.

Dhanak, M. R. and B. D. Bernardinis (1981). "The evolution of an elliptic vortex ring." In: *Journal of Fluid Mechanics* 109, pp. 189–216. DOI: 10.1017/S0022112081001006.

Domenici, P. and R. W. Blake (1997). "The kinematics and performance of fish fast-start swimming." In: *Journal of Experimental Biology* 200.8, pp. 1165–1178.

Domingues, M. O., S. M. Gomes, O. Roussel, and K. Schneider (2008). "An adaptive multiresolution scheme with local time stepping for evolutionary PDEs." In: *Journal of Computational Physics* 227.8, pp. 3758–3780. DOI: 10.1016/j.jcp.2007.11.046.

Domingues, M. O., S. M. Gomes, O. Roussel, and K. Schneider (2009). "Space-time adaptive multiresolution methods for hyperbolic conservation laws: Applications to compressible Euler equations." In: *Applied Numerical Mathematics* 59.9, pp. 2303–2321. DOI: 10.1016/j.apnum.2008.12.018.

Dong, H., M. Bozkurttas, R. Mittal, P. Madden, and G. V. Lauder (2010). "Computational modelling and analysis of the hydrodynamics of a highly deformable fish pectoral fin." In: *Journal of Fluid Mechanics* 645, pp. 345–373. DOI: 10.1017/S0022112009992941.

Donoho, D. L. (1992). *Interpolating Wavelet Transforms*. Tech. rep. Stanford University.

Dubuc, S. (1986). "Interpolation through an iterative scheme." In: *Journal of Mathematical Analysis and Applications* 114.1, pp. 185–204. DOI: 10.1016/0022-247X(86)90077-6.

Eloy, C. (2013). "On the best design for undulatory swimming." In: *Journal of Fluid Mechanics* 717, pp. 48–89. DOI: 10.1017/jfm.2012.561.

Epps, B. P. and A. H. Techet (2007). "Impulse generated during unsteady maneuvering of swimming fish." In: *Experiments in Fluids* 43.5, pp. 691–700. DOI: 10.1007/s00348-007-0401-4.

Fohl, T. and J. S. Turner (1975). "Colliding vortex rings." In: *Physics of Fluids* 18.4, p. 433. DOI: 10.1063/1.861169.

Fontaine, E., D. Lentink, S. Kranenbarg, U. K. Müller, J. L. van Leeuwen, A. H. Barr, and J. W. Burdick (2008). "Automated visual tracking for studying the ontogeny of zebrafish swimming." In: *Journal of Experimental Biology* 211.8, pp. 1305–1316. DOI: 10.1242/jeb.010272.

Frigo, M. and S. Johnson (1998). "FFTW: an adaptive software architecture for the FFT." In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)* 3, pp. 1381–1384. DOI: 10.1109/ICASSP.1998.681704.

Gazzola, M., M. Argentina, and L. Mahadevan (2014). "Scaling macroscopic aquatic locomotion." In: *Nature Physics* 10.10, pp. 758–761. DOI: 10.1038/nphys3078.

Gazzola, M., P. Chatelain, W. M. van Rees, and P. Koumoutsakos (2011). "Simulations of single and multiple swimmers with non-divergence free deforming geometries." In: *Journal of Computational Physics* 230.19, pp. 7093–7114. DOI: 10.1016/j.jcp.2011.04.025.

Gazzola, M., B. Hejazialhosseini, and P. Koumoutsakos (2014). "Reinforcement Learning and Wavelet Adapted Vortex Methods for Simulations of Self-propelled Swimmers." In: *SIAM Journal on Scientific Computing* 36.3, B622–B639. DOI: 10.1137/130943078.

Gazzola, M., C. Mimeau, A. A. Tchieu, and P. Koumoutsakos (2012). "Flow mediated interactions between two cylinders at finite Re numbers." In: *Physics of Fluids* 24.4, p. 43103. DOI: 10.1063/1.4704195.

Gazzola, M., W. M. van Rees, and P. Koumoutsakos (2012). "C-start: optimal start of larval fish." In: *Journal of Fluid Mechanics* 698, pp. 5–18. DOI: 10.1017/jfm.2011.558.

Gazzola, M., O. V. Vasilyev, and P. Koumoutsakos (2011). "Shape optimization for drag reduction in linked bodies using evolution strategies." In: *Computers & Structures* 89.11-12, pp. 1224–1231. DOI: 10.1016/j.compstruc.2010.09.001.

Gilmanov, A. and F. Sotiropoulos (2005). “A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies.” In: *Journal of Computational Physics* 207.2, pp. 457–492. DOI: 10.1016/j.jcp.2005.01.020.

Gray, J. (1933). “Studies in animal locomotion I. The movement of fish with special reference to the eel.” In: *Journal of Experimental Biology* 10.1, 88–U8.

Greengard, L. and V. Rokhlin (1987). “A fast algorithm for particle simulations.” In: *Journal of Computational Physics* 73.2, pp. 325–348. DOI: 10.1016/0021-9991(87)90140-9.

Griffiths, G., ed. (2003). *Technology and Applications of Autonomous Underwater Vehicles*. Ocean Science and Technology. Taylor and Francis, Abingdon, UK.

Haller, G. and G. Yuan (2000). “Lagrangian coherent structures and mixing in two-dimensional turbulence.” In: *Physica D: Nonlinear Phenomena* 147.3-4, pp. 352–370. DOI: 10.1016/S0167-2789(00)00142-1.

Hansen, N., S. D. Müller, and P. Koumoutsakos (2003). “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES).” In: *Evolutionary computation* 11.1, pp. 1–18. DOI: 10.1162/106365603321828970.

Hansen, N. and A. Ostermeier (2001). “Completely derandomized self-adaptation in evolution strategies.” In: *Evolutionary computation* 9.2, pp. 159–195. DOI: 10.1162/106365601750190398.

Hejazialhosseini, B., D. Rossinelli, M. Bergdorf, and P. Koumoutsakos (2010). “High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions.” In: *Journal of Computational Physics* 229.22, pp. 8364–8383. DOI: 10.1016/j.jcp.2010.07.021.

Hejazialhosseini, B., D. Rossinelli, C. Conti, and P. Koumoutsakos (2012). “High throughput software for direct numerical simulations of compressible two-phase flows.” In: *2012 International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, pp. 1–12. ISBN: 978-1-4673-0806-9. DOI: 10.1109/SC.2012.66.

Hejlesen, M. M., P. Koumoutsakos, A. Leonard, and J. H. Walther (2015). “Iterative Brinkman penalization for remeshed vortex methods.” In: *Journal of Computational Physics* 280, pp. 547–562. DOI: 10.1016/j.jcp.2014.09.029.

Hejlesen, M. M., J. T. Rasmussen, P. Chatelain, and J. H. Walther (2013). “A high order solver for the unbounded Poisson equation.” In: *Journal of Computational Physics* 252, pp. 458–467. DOI: 10.1016/j.jcp.2013.05.050.

Hertel, H. (1966). *Structure, form, movement*. Reinhold, New York.

Hieber, S. and P. Koumoutsakos (2008). “An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers.” In: *Journal of Computational Physics* 227.19, pp. 8636–8654. DOI: 10.1016/j.jcp.2008.06.017.

Hockney, R. W. and J. W. Eastwood (1981). *Computer simulation using particles*. McGraw-Hill.

Hou, T. Y. and Z. Lei (2009). “On the stabilizing effect of convection in three-dimensional incompressible flows.” In: *Communications on Pure and Applied Mathematics* 62.4, pp. 501–564. DOI: 10.1002/cpa.20254.

Hou, T. Y. and R. Li (2007). “Computing nearly singular solutions using pseudo-spectral methods.” In: *Journal of Computational Physics* 226.1, pp. 379–397. DOI: 10.1016/j.jcp.2007.04.014.

Hou, T. Y. and R. Li (2008). “Blowup or no blowup? The interplay between theory and numerics.” In: *Physica D: Nonlinear Phenomena* 237.14-17, pp. 1937–1944. DOI: 10.1016/j.physd.2008.01.018.

Howland, H. C. (1974). “Optimal strategies for predator avoidance: The relative importance of speed and manoeuvrability.” In: *Journal of Theoretical Biology* 47.2, pp. 333–350. DOI: 10.1016/0022-5193(74)90202-1.

Hu, W., Y. Yu, B. Tong, and H. Liu (2004). “A numerical and analytical study on a tail-flapping model for fish fast C-start.” In: *Acta Mechanica Sinica* 20.1, pp. 16–23. DOI: 10.1007/BF02484240.

Hunt, J. C. R., A. A. Wray, and P. Moin (1988). “Eddies, streams, and convergence zones in turbulent flows.” In: *Proceedings of the 1988 Summer Program, Center of Turbulence Research*. CTR-S88, pp. 193–208.

Hussain, F. (1983). “Coherent structures — reality and myth.” In: *Physics of Fluids* 26.10, p. 2816. DOI: 10.1063/1.864048.

Hussain, F. and K. Duraisamy (2011). “Mechanics of viscous vortex reconnection.” In: *Physics of Fluids* 23.2, p. 021701. DOI: 10.1063/1.3532039.

Ijspeert, A. J., A. Crespi, D. Ryczko, and J.-M. Cabelguen (2007). “From swimming to walking with a salamander robot driven by a spinal cord model.” In: *Science* 315.5817, pp. 1416–1420. DOI: 10.1126/science.1138353.

Jacobs, E. N., K. E. Ward, and R. M. Pinkerton (1933). *The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel*. Tech. rep. 460. NACA.

Jeong, J. and F. Hussain (1995). “On the identification of a vortex.” In: *Journal of Fluid Mechanics* 285, pp. 69–94. DOI: 10.1017/S0022112095000462.

Johnson, T. A. and V. C. Patel (1999). “Flow past a sphere up to a Reynolds number of 300.” In: *Journal of Fluid Mechanics* 378, pp. 19–70. DOI: 10.1017/S0022112098003206.

Kambe, T. and T. Minota (1983). “Acoustic Wave Radiated by Head-On Collision of Two Vortex Rings.” In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 386.1791, pp. 277–308. DOI: 10.1098/rspa.1983.0037.

Katumata, Y., U. K. Müller, and H. Liu (2009). “Computation of Self-Propelled Swimming in Larva Fishes.” In: *Journal of Biomechanical Science and Engineering* 4.1, pp. 54–66. DOI: 10.1299/jbse.4.54.

Kern, S., N. Hansen, and P. Koumoutsakos (2006). “Local meta-models for optimization using evolution strategies.” In: *Parallel Problem Solving from Nature - PPSN IX*. Ed. by T. P. Runarsson, H. G. Beyer, E. Burke, J. J. MereloGuervos, L. D. Whitley, and X. Yao. Vol. 4193. Lecture Notes in Computer Science. Springer-Verlag Berlin, pp. 939–948. DOI: 10.1007/11844297.

Kern, S. and P. Koumoutsakos (2006). “Simulations of optimized anguilliform swimming.” In: *Journal of Experimental Biology* 209.24, pp. 4841–4857. DOI: 10.1242/jeb.02526.

Kerr, R. M. (1993). “Evidence for a singularity of the three-dimensional, incompressible Euler equations.” In: *Physics of Fluids A: Fluid Dynamics* 5.7, p. 1725. DOI: 10.1063/1.858849.

Kerr, R. M. and F. Hussain (1989). “Simulation of vortex reconnection.” In: *Physica D: Nonlinear Phenomena* 37.1-3, pp. 474–484. DOI: 10.1016/0167-2789(89)90151-6.

Kida, S. (1994). “Vortex reconnection.” In: *Annual Review of Fluid Mechanics* 26, pp. 169–189.

Kida, S. and M. Takaoka (1988). “Reconnection of vortex tubes.” In: *Fluid Dynamics Research* 3.1-4, pp. 257–261. DOI: 10.1016/0169-5983(88)90075-5.

Kida, S., M. Takaoka, and F. Hussain (1989). “Reconnection of two vortex rings.” In: *Physics of Fluids A: Fluid Dynamics* 1.4, pp. 630–632. DOI: 10.1063/1.857436.

Kida, S., M. Takaoka, and F. Hussain (1991). “Collision of two vortex rings.” In: *Journal of Fluid Mechanics* 230, pp. 583–646. DOI: 10.1017/S0022112091000903.

Kleppmann, M. (2007). *Simulation of colliding constrained rigid bodies*. Tech. rep. University of Cambridge.

Kohannim, S. and T. Iwasaki (2014). "Analytical insights into optimality and resonance in fish swimming." In: *Journal of the Royal Society, Interface / the Royal Society* 11.92, p. 20131073. DOI: 10.1098/rsif.2013.1073.

Kolmogorov, A. N. (1991). "On the Degeneration of Isotropic Turbulence in an Incompressible Viscous Fluid." In: *Selected Works of A. N. Kolmogorov* 33.6. Ed. by V. M. Tikhomirov, pp. 319–323. DOI: 10.1007/978-94-011-3030-1.

Koumoutsakos, P. (1997). "Inviscid Axisymmetrization of an Elliptical Vortex." In: *Journal of Computational Physics* 138.2, pp. 821–857. DOI: 10.1006/jcph.1997.5749.

Koumoutsakos, P. (2005). "Multiscale Flow Simulations Using Particles." In: *Annual Review of Fluid Mechanics* 37.1, pp. 457–487. DOI: 10.1146/annurev.fluid.37.061903.175753.

Koumoutsakos, P. and A. Leonard (1995). "High-resolution simulations of the flow around an impulsively started cylinder using vortex methods." In: *Journal of Fluid Mechanics* 296, pp. 1–38. DOI: 10.1017/S0022112095002059.

Lauder, G. V. (2000). "Function of the caudal fin during locomotion in fishes: kinematics, flow visualization and evolutionary patterns." In: *American Zoology* 40.1, pp. 101–122. DOI: 10.1093/icb/40.1.101.

Lauder, G. V., E. J. Anderson, J. Tangorra, and P. G. A. Madden (2007). "Fish biorobotics: kinematics and hydrodynamics of self-propulsion." In: *Journal of Experimental Biology* 210.16, pp. 2767–2780. DOI: 10.1242/jeb.000265.

Leonard, A. (1980). "Vortex methods for flow simulation." In: *Journal of Computational Physics* 37.3, pp. 289–335. DOI: 10.1016/0021-9991(80)90040-6.

Leonard, A. (1985). "Computing Three-Dimensional Incompressible Flows with Vortex Elements." In: *Annual Review of Fluid Mechanics* 17.1, pp. 523–559. DOI: 10.1146/annurev.fl.17.010185.002515.

Lesieur, M. (2008). *Turbulence in Fluids*. 4th. Springer, Dordrecht.

Lim, T. T. and T. B. Nickels (1992). "Instability and reconnection in the head-on collision of two vortex rings." In: *Nature* 357.6375, pp. 225–227. DOI: 10.1038/357225a0.

Magni, A. and G.-H. Cottet (2012). “Accurate, non-oscillatory, remeshing schemes for particle methods.” In: *Journal of Computational Physics* 231.1, pp. 152–172. DOI: 10.1016/j.jcp.2011.09.005.

Mallat, S. (1989). “A theory for multiresolution signal decomposition: the wavelet representation.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7, pp. 674–693. DOI: 10.1109/34.192463.

Marshall, J. S., P. Brancher, and A. Giovannini (2001). “Interaction of unequal anti-parallel vortex tubes.” In: *Journal of Fluid Mechanics* 446, pp. 229–252.

Martin, D. F. and P. Colella (2000). “A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations.” In: *Journal of Computational Physics* 163.2, pp. 271–312. DOI: 10.1006/jcph.2000.6575.

Melander, M. V. and F. Hussain (1988). “Cut-and-connect of two antiparallel vortex tubes.” In: *Proceedings of the 1988 Summer Program, Center of Turbulence Research*, pp. 257–286.

Melander, M. V. and F. Hussain (1993). “Polarized vorticity dynamics on a vortex column.” In: *Physics of Fluids A: Fluid Dynamics* 5.8, p. 1992. DOI: 10.1063/1.858826.

Melander, M. V. and N. J. Zabusky (1988). “Interaction and "apparent" reconnection of 3D vortex tubes via direct numerical simulations.” In: *Fluid Dynamics Research* 3.1-4, pp. 247–250. DOI: 10.1016/0169-5983(88)90073-1.

Miniati, F. and P. Colella (2007). “Block structured adaptive mesh and time refinement for hybrid, hyperbolic+N-body systems.” In: *Journal of Computational Physics* 227.1, pp. 400–430. DOI: 10.1016/j.jcp.2007.07.035.

Mininni, P. D. and A. Pouquet (2010). “Rotating helical turbulence. I. Global evolution and spectral behavior.” In: *Physics of Fluids* 22.3, p. 035105. DOI: 10.1063/1.3358466.

Mittal, R., H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. von Loebbecke (2008). “A Versatile Sharp Interface Immersed Boundary Method For Incompressible Flows With Complex Boundaries.” In: *Journal of Computational Physics* 227.10, pp. 4825–4852. DOI: 10.1016/j.jcp.2008.01.028.

Monaghan, J. (1985). “Extrapolating B splines for interpolation.” In: *Journal of Computational Physics* 60.2, pp. 253–262. DOI: 10.1016/0021-9991(85)90006-3.

Moses, H. E. (1971). “Eigenfunctions of the Curl Operator, Rotationally Invariant Helmholtz Theorem, and Applications to Electromagnetic Theory and Fluid Mechanics.” In: *SIAM Journal on Applied Mathematics* 21.1, pp. 114–144. DOI: 10.1137/0121015.

Moshkin, N. P. and P. Suwannasri (2012). “Two regimes of self-propelled motion of a torus rotating about its centerline in a viscous incompressible fluid at intermediate Reynolds numbers.” In: *Physics of Fluids* 24.5, p. 053603. DOI: 10.1063/1.4717760.

Müller, U. K. (2004). “Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development.” In: *Journal of Experimental Biology* 207.5, pp. 853–868. DOI: 10.1242/jeb.00821.

Müller, U. K., J. G. M. van den Boogaart, and J. L. van Leeuwen (2008). “Flow patterns of larval fish: undulatory swimming in the intermediate flow regime.” In: *Journal of Experimental Biology* 211.2, pp. 196–205. DOI: 10.1242/jeb.005629.

Nachtigall, W. and K. Bluechel (2000). *Das grosse Buch der Bionik*. Deutsche Verlags-Anstalt.

Naitoh, T., N. Fukuda, T. Gotoh, H. Yamada, and K. Nakajima (2002). “Experimental study of axial flow in a vortex ring.” In: *Physics of Fluids* 14.1, pp. 143–149. DOI: 10.1063/1.1420745.

Nawroth, J. C., H. Lee, A. W. Feinberg, C. M. Ripplinger, M. L. McCain, A. Grosberg, J. O. Dabiri, and K. K. Parker (2012). “A tissue-engineered jellyfish with biomimetic propulsion.” In: *Nature biotechnology* 30.8, pp. 792–797. DOI: 10.1038/nbt.2269.

Ortega, J. M., R. L. Bristol, and Ö. Savas (2003). “Experimental study of the instability of unequal-strength counter-rotating vortex pairs.” In: *Journal of Fluid Mechanics* 474, pp. 35–84. DOI: 10.1017/S0022112002002446.

Oshima, Y. (1978). “Head-on Collision of Two Vortex Rings.” In: *Journal of the Physical Society of Japan* 44.1, pp. 328–331. DOI: 10.1143/JPSJ.44.328.

Oshima, Y. and S. Asaka (1975). *Interaction of two vortex rings moving side by side*. Nature Science Report 1. Ochanomizu University.

Oshima, Y. and N. Izutsu (1988). “Cross-linking of two vortex rings.” In: *Physics of Fluids* 31.9, pp. 2401–2403. DOI: 10.1063/1.866593.

Parichy, D. M., M. R. Elizondo, M. G. Mills, T. N. Gordon, and R. E. Engeszer (2009). “Normal table of postembryonic zebrafish development: staging by externally visible anatomy of the living fish.” In: *Developmental dynamics : an official publication of the American Association of Anatomists* 238.12, pp. 2975–3015. DOI: 10.1002/dvdy.22113.

Patankar, N. A. and N. Sharma (2005). “A fast projection scheme for the direct numerical simulation of rigid particulate flows.” In: *Communications in Numerical Methods in Engineering* 21.8, pp. 419–432. DOI: 10.1002/cnm.756.

Peskin, C. S. (1982). “The Fluid Dynamics of Heart Valves: Experimental, Theoretical, and Computational Methods.” In: *Annual Review of Fluid Mechanics* 14.1, pp. 235–259. DOI: 10.1146/annurev.fl.14.010182.001315.

Ploumhans, P. and G. Winckelmans (2000). “Vortex Methods for High-Resolution Simulations of Viscous Flow Past Bluff Bodies of General Geometry.” In: *Journal of Computational Physics* 165.2, pp. 354–406. DOI: 10.1006/jcph.2000.6614.

Pumir, A. and R. M. Kerr (1987). “Numerical simulation of interacting vortex tubes.” In: *Physical Review Letters* 58.16, pp. 1636–1639. DOI: 10.1103/PhysRevLett.58.1636.

Pumir, A. and E. D. Siggia (1987). “Vortex dynamics and the existence of solutions to the Navier-Stokes equations.” In: *Physics of Fluids* 30.6, pp. 1606–1626. DOI: 10.1063/1.866226.

Rahimian, A., I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros (2010). “Petascale Direct Numerical Simulation of Blood Flow on 200K Cores and Heterogeneous Architectures.” In: *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, pp. 1–11. ISBN: 978-1-4244-7557-5. DOI: 10.1109/SC.2010.42.

Raviart, P. A. (1985). “An analysis of particle methods.” In: *Numerical methods in fluid dynamics*. Ed. by F. Brezzi. Vol. 1127. Lecture Notes in Mathematics. Springer Berlin Heidelberg, pp. 243–324. DOI: 10.1007/BFb0074532.

Rechenberg, I. (1994). *Evolutionsstrategie ’94*. frommann-holzboog.

Rees, W. M. van, M. Gazzola, and P. Koumoutsakos (2013). “Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers.” In: *Journal of Fluid Mechanics* 722, R3. DOI: 10.1017/jfm.2013.157.

Rees, W. M. van, F. Hussain, and P. Koumoutsakos (2012). “Vortex tube reconnection at $Re = 10^4$.” In: *Physics of Fluids* 24.7, p. 75105. DOI: 10.1063/1.4731809.

Rees, W. M. van, A. Leonard, D. Pullin, and P. Koumoutsakos (2011). “A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers.” In: *Journal of Computational Physics* 230.8, pp. 2794–2805. DOI: 10.1016/j.jcp.2010.11.031.

Rees, W. M. van, D. Rossinelli, P. Hadjidoukas, and P. Koumoutsakos (2014). “High performance CPU/GPU multiresolution Poisson solver.” In: *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*. Ed. by M. Bader, A. Bode, H.-J. Bungartz,

M. Gerndt, G. R. Joubert, and F. Peters. IOS Press, pp. 481–490. DOI: 10.3233/978-1-61499-381-0-481.

Reid, D. A. P., H. Hildenbrandt, J. T. Padding, and C. K. Hemelrijk (2012). “Fluid dynamics of moving fish in a two-dimensional multiparticle collision dynamics model.” In: *Physical Review E* 85.2, p. 021901. DOI: 10.1103/PhysRevE.85.021901.

Robison, A., M. Voss, and A. Kukanov (2008). “Optimization via Reflection on Work Stealing in TBB.” In: *2008 IEEE International Symposium on Parallel and Distributed Processing*. IEEE, pp. 1–8. ISBN: 978-1-4244-1693-6. DOI: 10.1109/IPDPS.2008.4536188.

Rohr, J. J. and F. Fish (2004). “Strouhal numbers and optimization of swimming by odontocete cetaceans.” In: *Journal of Experimental Biology* 207.10, pp. 1633–1642. DOI: 10.1242/jeb.00948.

Rosenhead, L. (1931). “The Formation of Vortices from a Surface of Discontinuity.” In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 134.823, pp. 170–192. DOI: 10.1098/rspa.1931.0189.

Rossinelli, D., B. Hejazialhosseini, W. M. van Rees, M. Gazzola, M. Bergdorf, and P. Koumoutsakos (2015). “MRAG-I2D: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures.” In: *Journal of Computational Physics* 288, pp. 1–18. DOI: 10.1016/j.jcp.2015.01.035.

Rossinelli, D. (2011). “Multiresolution flow simulations on multi/many-core architectures.” Doctoral dissertation. ETH Zurich. DOI: 10.3929/ethz-a-006483930.

Rossinelli, D., B. Hejazialhosseini, M. Bergdorf, and P. Koumoutsakos (2011). “Wavelet-adaptive solvers on multi-core architectures for the simulation of complex systems.” In: *Concurrency and Computation: Practice and Experience* 23.2, pp. 172–186. DOI: 10.1002/cpe.1639.

Rossinelli, D., B. Hejazialhosseini, D. G. Spampinato, and P. Koumoutsakos (2011). “Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids.” In: *SIAM Journal on Scientific Computing* 33.2, pp. 512–540. DOI: 10.1137/100795930.

Roussel, O., K. Schneider, A. Tsigulin, and H. Bockhorn (2003). “A conservative fully adaptive multiresolution algorithm for parabolic PDEs.” In: *Journal of Computational Physics* 188.2, pp. 493–523. DOI: 10.1016/S0021-9991(03)00189-X.

Saffman, P. G. (1990). “A model of vortex reconnection.” In: *Journal of Fluid Mechanics* 212, pp. 395–402. DOI: 10.1017/S0022112090002026.

Sbalzarini, I., J. H. Walther, M. Bergdorf, S. Hieber, E. Kotsalis, and P. Koumoutsakos (2006). “PPM – A highly efficient parallel particle-mesh library for the simulation of continuum systems.” In: *Journal of Computational Physics* 215.2, pp. 566–588. DOI: 10.1016/j.jcp.2005.11.017.

Schatzle, P. R. (1987). “An experimental study of fusion of vortex rings.” Doctoral dissertation. California Institute of Technology, Pasadena, California.

Schneider, K. and M. Farge (2002). “Adaptive Wavelet Simulation of a Flow around an Impulsively Started Cylinder Using Penalisation.” In: *Applied and Computational Harmonic Analysis* 12.3, pp. 374–380. DOI: 10.1006/acha.2002.0378.

Schneider, K. and O. V. Vasilyev (2010). “Wavelet Methods in Computational Fluid Dynamics.” In: *Annual Review of Fluid Mechanics* 42.1, pp. 473–503. DOI: 10.1146/annurev-fluid-121108-145637.

Schwefel, H. P. (1975). “Evolutionsstrategie und numerische Optimierung.” Doctoral dissertation. TU Berlin.

Scorer, R. S. and L. J. Davenport (1970). “Contrails and aircraft downwash.” In: *Journal of Fluid Mechanics* 43.03, pp. 451–464. DOI: 10.1017/S0022112070002501.

Shelley, M. J., D. I. Meiron, and S. A. Orszag (1993). “Dynamical aspects of vortex reconnection of perturbed anti-parallel vortex tubes.” In: *Journal of Fluid Mechanics* 246, pp. 613–652. DOI: 10.1017/S0022112093000291.

Shirgaonkar, A. A., M. A. MacIver, and N. A. Patankar (2009). “A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion.” In: *Journal of Computational Physics* 228.7, pp. 2366–2390. DOI: 10.1016/j.jcp.2008.12.006.

Stanaway, S., K. Shariff, and F. Hussain (1988). “Head-on collision of viscous vortex rings.” In: *Proceedings of the 1988 Summer Program, Center of Turbulence Research*, pp. 287–309.

Sugiyama, K., S. Ii, S. Takeuchi, S. Takagi, and Y. Matsumoto (2011). “A full Eulerian finite difference approach for solving fluid-structure coupling problems.” In: *Journal of Computational Physics* 230.3, pp. 596–627. DOI: 10.1016/j.jcp.2010.09.032.

Sungnul, S. and N. P. Moshkin (2009). “Effect of Rotation Rates and Gap Spacing on the Structure of Low Reynolds Number Flow over Two Rotating Circular Cylinders.” In:

Computational Fluid Dynamics 2008. Ed. by H. Choi, H. G. Choi, and J. Y. Yoo. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 771–777. ISBN: 978-3-642-01272-3. DOI: 10.1007/978-3-642-01273-0.

Sutton, S. R. and A. G. Barto (1998). *Reinforcement Learning: An Introduction*. The MIT Press Cambridge, Massachusetts The MIT Press, Cambridge, Massachussets, London, England.

Suwannasri, P. (2013). "Numerical simulation of self-propelled motion of two rotating side-by-side circular cylinders." In: *Southeast-Asian Journal of Sciences* 2.1, pp. 19–26.

Taylor, G. K., R. L. Nudds, and A. L. R. Thomas (2003). "Flying and swimming animals cruise at a Strouhal number tuned for high power efficiency." In: *Nature* 425.6959, pp. 707–711. DOI: 10.1038/nature02000.

Tokic, G. and D. K. P. Yue (2012). "Optimal shape and motion of undulatory swimming organisms." In: *Proceedings of the Royal Society B: Biological Sciences* 279.1740, pp. 3065–3074. DOI: 10.1098/rspb.2012.0057.

Tomboulides, A. G. and S. A. Orszag (2000). "Numerical investigation of transitional and weak turbulent flow past a sphere." In: *Journal of Fluid Mechanics* 416, pp. 45–73. DOI: 10.1017/S0022112000008880.

Towers, J. D. (2009). "Finite difference methods for approximating Heaviside functions." In: *Journal of Computational Physics* 228.9, pp. 3478–3489. DOI: 10.1016/j.jcp.2009.01.026.

Triantafyllou, G. S., M. S. Triantafyllou, and M. Grosenbaugh (1993). "Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion." In: *Journal of Fluids and Structures* 7.2, pp. 205–224. DOI: 10.1006/jfls.1993.1012.

Triantafyllou, M. S., F. S. Hover, A. H. Techet, and D. K. P. Yue (2005). "Review of Hydrodynamic Scaling Laws in Aquatic Locomotion and Fishlike Swimming." In: *Applied Mechanics Reviews* 58.4, pp. 226–237. DOI: 10.1115/1.1943433.

Triantafyllou, M. S. and G. S. Triantafyllou (1995). "An Efficient Swimming Machine." In: *Scientific American* 272.3, pp. 64–70. DOI: 10.1038/scientificamerican0395-64.

Triantafyllou, M. S., G. S. Triantafyllou, and D. K. P. Yue (2000). "Hydrodynamics of Fishlike Swimming." In: *Annual Review of Fluid Mechanics* 32.1, pp. 33–53. DOI: 10.1146/annurev.fluid.32.1.33.

Tytell, E. D. (2004a). "The hydrodynamics of eel swimming: I. Wake structure." In: *Journal of Experimental Biology* 207.11, pp. 1825–1841. DOI: 10.1242/jeb.00968.

Tytell, E. D. (2004b). "The hydrodynamics of eel swimming II. Effect of swimming speed." In: *Journal of Experimental Biology* 207.19, pp. 3265–3279. DOI: 10.1242/jeb.01139.

Tytell, E. D., I. Borazjani, F. Sotiropoulos, T. V. Baker, E. J. Anderson, and G. V. Lauder (2010). "Disentangling the functional roles of morphology and motion in the swimming of fish." In: *Integrative and comparative biology* 50.6, pp. 1140–1154. DOI: 10.1093/icb/icq057.

Tytell, E. D. and G. V. Lauder (2002). "The C-start escape response of Polypterus senegalus: bilateral muscle activity and variation during stage 1 and 2." In: *Journal of Experimental Biology* 205.17, pp. 2591–2603.

Vasilyev, O. V. and N. K.-R. Kevlahan (2002). "Hybrid wavelet collocation-Brinkman penalization method for complex geometry flows." In: *International Journal for Numerical Methods in Fluids* 40.3-4, pp. 531–538. DOI: 10.1002/fld.307.

Virk, D., F. Hussain, and R. M. Kerr (1995). "Compressible vortex reconnection." In: *Journal of Fluid Mechanics* 304, pp. 47–86. DOI: 10.1017/S0022112095004344.

Virk, D., M. V. Melander, and F. Hussain (1994). "Dynamics of a polarized vortex ring." In: *Journal of Fluid Mechanics* 260, pp. 23–55. DOI: 10.1017/S0022112094003423.

Walker, J. A., C. K. Ghalambor, O. L. Griset, D. Mckenney, and D. N. Reznick (2005). "Do faster starts increase the probability of evading predators?" In: *Functional Ecology* 19.5, pp. 808–815. DOI: 10.1111/j.1365-2435.2005.01033.x.

Webb, P. W. (1984). "Body Form, Locomotion and Foraging in Aquatic Vertebrates." In: *American Zoology* 24.1, pp. 107–120. DOI: 10.1093/icb/24.1.107.

Wee, D. and A. F. Ghoniem (2006). "Modified interpolation kernels for treating diffusion and remeshing in vortex methods." In: *Journal of Computational Physics* 213.1, pp. 239–263. DOI: 10.1016/j.jcp.2005.08.009.

Weihs, D. (1973a). "Hydromechanics of Fish Schooling." In: *Nature* 241.5387, pp. 290–291. DOI: 10.1038/241290a0.

Weihs, D. (1973b). "The mechanism of rapid starting of slender fish." In: *Biorheology* 10, pp. 343–350.

Weihs, D. and P. W. Webb (1984). “Optimal avoidance and evasion tactics in predator-prey interactions.” In: *Journal of Theoretical Biology* 106.2, pp. 189–206. DOI: 10.1016/0022-5193(84)90019-5.

Whittlesey, R. W., S. Liska, and J. O. Dabiri (2010). “Fish schooling as a basis for vertical axis wind turbine farm design.” In: *Bioinspiration & biomimetics* 5.3, p. 035005. DOI: 10.1088/1748-3182/5/3/035005.

Wilga, C. D. and G. V. Lauder (2004). “Biomechanics: hydrodynamic function of the shark’s tail.” In: *Nature* 430.7002, p. 850. DOI: 10.1038/430850a.

Williamson, C. H. K. and A. Roshko (1988). “Vortex formation in the wake of an oscillating cylinder.” In: *Journal of Fluids and Structures* 2.4, pp. 355–381. DOI: 10.1016/S0889-9746(88)90058-8.

Williamson, J. (1980). “Low-storage Runge-Kutta schemes.” In: *Journal of Computational Physics* 35.1, pp. 48–56. DOI: 10.1016/0021-9991(80)90033-9.

Winckelmans, G. (2004). “Vortex Methods.” In: *Encyclopedia of Computational Mechanics*. Ed. by E. Stein, R. de Borst, and T. J. R. Hughes. Vol. 3. Chichester: John Wiley & Sons, Ltd. ISBN: 0470091355. DOI: 10.1002/0470091355.

Winckelmans, G. and A. Leonard (1993). “Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows.” In: *Journal of Computational Physics* 109.2, pp. 247–273. DOI: 10.1006/jcph.1993.1216.

Wu, T. Y. (2011). “Fish Swimming and Bird/Insect Flight.” In: *Annual Review of Fluid Mechanics* 43.1, pp. 25–58. DOI: 10.1146/annurev-fluid-122109-160648.

Yokota, R. and L. Barba (2012). “Hierarchical N-body Simulations with Autotuning for Heterogeneous Systems.” In: *Computing in Science & Engineering* 14.3, pp. 30–39. DOI: 10.1109/MCSE.2012.1.

Zabusky, N. J., O. Boratav, R. Pelz, M. Gao, D. Silver, and S. Cooper (1991). “Emergence of coherent patterns of vortex stretching during reconnection: A scattering paradigm.” In: *Physical Review Letters* 67.18, pp. 2469–2472. DOI: 10.1103/PhysRevLett.67.2469.

Zabusky, N. J. and M. V. Melander (1989). “Three-dimensional vortex tube reconnection: Morphology for orthogonally-offset tubes.” In: *Physica D: Nonlinear Phenomena* 37.1-3, pp. 555–562. DOI: 10.1016/0167-2789(89)90160-7.

Copyright disclaimer

- Chapter 3 is partially reprinted from van Rees, Leonard, et al. (2011)
- Chapter 4 is partially reprinted from van Rees, Hussain, and Koumoutsakos (2012)
- Chapter 5 is partially reprinted from Gazzola, Chatelain, et al. (2011)
- Chapter 6 is partially reprinted from Gazzola, van Rees, and Koumoutsakos (2012)
- Chapter 7 is partially reprinted from van Rees, Gazzola, and Koumoutsakos (2013)
- Chapter 8 is unpublished work printed here with kind permission of its co-authors Mattia Gazzola and Petros Koumoutsakos
- Chapter 9 is partially reprinted from Rossinelli, Hejazialhosseini, van Rees, et al. (2015)
- Chapter 10 is partially reprinted from van Rees, Rossinelli, et al. (2014)
- Chapter 11 is unpublished work printed here with kind permission of its co-authors Guido Novati and Petros Koumoutsakos
- Chapter 12 is unpublished work printed here with kind permission of its co-authors Diego Rossinelli and Petros Koumoutsakos

Own Publications

- W. M. van Rees, A. Leonard, D. I. Pullin, and P. Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *Journal of Computational Physics*, Vol. 230, pages 2794–2805, 2011.

- M. Gazzola, P. Chatelain, W. M. van Rees, and P. Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, Vol. 230, pages 7093–7114, 2011.

- M. Gazzola, W. M. van Rees, and P. Koumoutsakos. C-start: optimal start of larval fish. *Journal of Fluid Mechanics*, Vol. 698, pages 5–18, 2012 (**Cover article**).

- W. M. van Rees, F. Hussain, and P. Koumoutsakos. Vortex tube reconnection at Re$=10^4$. *Physics of Fluids*, Vol. 24, pages 075105-1–07105-14, 2012.

- W. M. van Rees, M. Gazzola, and P. Koumoutsakos. Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers. *Journal of Fluid Mechanics*, Vol. 722, pages R3-1–R3-12, 2013 (**Cover article**).

- W. M. van Rees, D. Rossinelli, P. Hadjidoukas, and P. Koumoutsakos. High performance CPU/GPU multiresolution Poisson solver. In *Advances in Parallel Computing*, Vol. 25, IOS Press, pages 481–490, 2014.

- D. Rossinelli, B. Hejazialhosseini, W. M. van Rees, M. Gazzola, M. Bergdorf, and P. Koumoutsakos. MRAG-I2D: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *Journal of Computational Physics*, Vol. 288, pages 1–18, 2015.