Dissertation

For the Doctoral Degree of Engineering

Vortex Particle Redistribution and Regularisation

Matthias Kirchhart

August 2017

School of Science for Open and Environmental Systems
Graduate School of Science and Technology
Keio University

Matthias Kirchhart
Graduate School of Science and Technology
Keio University
3–14–1 Hiyoshi, Kōhoku-ku
Yokohama-shi 233–8522, Japan
kirchhart@keio.jp

To my cat, who loves me even when I am distant.

Abstract

Vortex methods are particle schemes that were first introduced as a tool to solve inviscid, unbounded flow problems in two- and three-dimensional space. In this thesis we briefly introduce the mathematical framework necessary to understand vortex methods and their benefits. When one tries to extend the method to viscous or bounded flows, however, one faces several additional problems. Our main contributions to the field address three of the problems encountered.

Our first contribution is a new scheme to handle unbounded, viscous flows. Based on the vorticity redistribution method, this scheme requires neither a frequent regridding of the particle field, nor makes use of the viscous splitting. It can thus be used with higher order time-stepping methods. Its consistency, stability, and conservation properties are proven. Together with the new heuristics of reduced operators and small neighbourhoods we demonstrate in numerical experiments that the method can be implemented efficiently on computers. The numerical results are in good agreement with the analysis.

As a second contribution we propose a new scheme to tackle the particle regularisation problem in bounded domains. This problem refers to the task of obtaining a smooth approximation of a function from a given particle field. To this end we construct a new class of globally smooth finite element spaces and prove their approximation properties. The global smoothness of these spaces will allow us to use them as test-functions for particle approximations. The regularisation problem is then modelled as a perturbation to a stabilised L^2-projection onto fictitious domains. After proving consistency, stability, and convergence of the method we show that optimal results are obtained when choosing $\sigma \sim \sqrt{h}$, where σ refers to the smoothing length and h to the particle spacing. As a consequence the complexity of the velocity computation can be reduced from $\mathcal{O}(h^{-d})$ to $\mathcal{O}(h^{-d/2})$. Numerical experiments confirm the analysis and that the derived error bounds are sharp.

Our third contribution are simple and efficient formulae to evaluate the Biot–Savart integral on three-dimensional domains using tetrahedral meshes. The derived formulae are exact for piecewise linear functions. Compared to the previously published formulae by Suh, the presented approach is numerically more stable and reduces the number of required arctangent evaluations from twelve to four. The increased stability is demonstrated in a simple numerical example.

We finish this thesis with concluding remarks on the results and possible interesting topics for future research.

Contents

Preface

When I first came in contact with vortex methods during my exchange semester at Keio University in 2012, I was immediately fascinated by the elegance of their underlying ideas. Vortex methods are numerical schemes for solving the incompressible Navier–Stokes equations. But instead of using their usual formulation in terms of velocity and pressure, vortex methods make use of their equivalent formulation in terms of vorticity. Using the vorticity as the primary variable comes with several benefits: the equations take the form of a transport equation from which the pressure variable and the solenoidal condition have been eliminated. By retrieving the velocity from the vorticity using the Biot–Savart law, this equation can be solved in a semi-analytical manner. This is done by discretising the vorticity field using 'particles' which are then convected with the flow. It is this natural treatment of convection which makes vortex methods so physically appealing and renders them essentially free of any artificial viscosity. This makes vortex methods ideal candidates for the direct numerical simulations of turbulence.

In many flows of practical interest, such as bluff body flows, the velocity field has global support. Conventional flow solvers based on the Navier–Stokes equations in their velocity–pressure formulation invariably need to artificially restrict the computational domain and have to introduce potentially non-physical boundary conditions. The vorticity, on the other hand, usually vanishes outside of a bounded region. Vortex methods make use of this fact, as only this bounded region needs to be discretised. The use of the Biot–Savart law then automatically guarantees that the far-field condition is fulfilled exactly.

Despite these apparent advantages vortex methods are not in widespread use today. The reason for this is twofold. The application of the Biot–Savart law to the particles leads to an N-body problem, for which the obvious direct summation approach leads to a computational complexity of $\mathcal{O}(N^2)$. This made the method prohibitively expensive for all but the smallest problems. Thus vortex methods remained an academic curiosity and the amount of research carried out on them was fairly limited. This changed with the development of fast summation techniques such as the Barnes–Hut treecode or Greengard and Rokhlin's fast multipole method in the late 1980ies, which addressed this problem in a satisfactory manner.

Vortex methods excel in the simulation of unbounded flows and in avoiding artificial viscosity; it is in this area that the theory behind vortex methods has reached its greatest maturity. On the other hand, the treatment of *physical* viscosity and solid boundaries remain problematic for both practical and theoretical reasons. Most commonly used viscous schemes rely on a frequent regridding of the particles and/or on the so-called viscous splitting. While regridding reintroduces artificial viscosity, the splitting limits

the method's accuracy in time to first order. In the treatment of solid boundaries one faces two problems: boundary conditions in terms of the vorticity are typically unavailable and need to be determined in a separate computation. But the perhaps even bigger problem lies in the need to regularise particles near boundaries, which has also been pointed out by Cottet and Koumoutsakos in the introduction of their monograph.

Vortex methods are intimately linked to the vorticity equation and make use of analytic solutions where they are available. This gives them an advantage compared to the commonly used general purpose methods. It is my belief that vortex methods thus have the potential to become a new standard tool in computational fluid dynamics, next to other well established methods such as finite volumes. But in order for this to happen, the aforementioned problems need to be solved in a theoretically as well as practically satisfactory manner first. During my time as a doctoral student I therefore focused on finding solutions for a small subset of the problems involved. It was my target to develop solutions which have both a solid mathematical foundation, while at the same time being practical and efficient to implement. This thesis is the result of this effort, and I hope that its contents will be useful in the future development of vortex methods.

Yokohama, August 2017.

Acknowledgements

I want to use this opportunity to express my deepest gratitude to my doctoral advisor, Professor Shinnosuke Obi. Without his support and advice this thesis would have been impossible. He also made me aware of the MEXT scholarship programme and the Keio Leading-edge Laboratory of Science and Technology, which funded my living and research expenses, respectively. My special thanks go to Professors Koji Fukagata, Keita Ando, and Rio Yokota. Their comments on my work were particularly helpful. My former colleague and friend Kentaro Kato helped me on many occasions, both in everyday life and research. The same holds true for my current colleagues Yūsuke Seike and Yūki Norimoto. I also owe my thanks to my previous advisors Sven Groß and Arnold Reusken, as well as Christoph Lehrenfeld and the remaining DROPS team of RWTH Aachen University, Germany. The knowledge about the extended finite element method (XFEM) gained during my time there would unexpectedly prove to be very useful in the field of vortex methods.

Chapter 1

A Brief History of Vortex Methods

Modern vortex methods and particle methods in general are unique in the sense that they use highly irregular objects—Dirac delta distributions—to approximate smooth flows. On first sight, this might seem unintuitive. Much to the contrary, however, particle methods predate the modern distribution theory and are in fact deeply based on physical intuition. Before moving on to the mathematical details in the next chapter, we therefore try to give an outline of the history of vortex methods here. It is our intention to highlight developments that we believe are of particular importance. Such a review is necessarily biased and incomplete, and we want to apologise to those authors we omitted. To at least partly accommodate for them we included several references to extensive reviews of the particular eras.

1.1 Origins (1858–1973)

Vortex methods were first proposed in 1931 by Louis Rosenhead to predict the unstable behaviour of vortex sheets in inviscid flows. In modern terms such flows correspond to solutions to the Euler equations whose vorticity is only defined in the sense of certain surface functionals, i. e., a special class of distributions. Similarly, the first vortex methods discretised such vortex sheets using point functionals; an even more irregular class of distributions. To understand the motivation behind the study of such irregular flows it is necessary to go back in time to the very beginning of the study of vortical flows. From a modern perspective it is interesting to study the old papers, as they treat highly irregular mathematical objects—distributions—with great elegance and physical intuition; objects that were only rigorously defined and formalised about a century later.

1.1.1 Advent of Vortex Dynamics

The beginning of the investigations on vortical flows can be pinpointed exactly: Hermann Helmholtz—who only later became to be known as Hermann *von* Helmholtz—published his landmark article on vortex motion in 1858 [1]. At this point the Euler equations had already been known for a century [2]. The influence of viscosity, however, had only recently been studied independently by Navier, Poisson, Saint-Venant, and Stokes [3]; the number of these independent works can be seen as an indicator that they were not

well known at the time. Unaware of these works, but well aware of the importance of friction, Helmholtz wanted to advance the field in the area [1, pp. 25f]:

> Der Einfluß der Reibung auf Flüssigkeiten konnte bisher noch nicht mathematisch definirt werden, und doch ist derselbe in allen Fällen, wo es sich nicht um unendlich kleine Schwingungen handelt, sehr groß, und bringt die bedeutendsten Abweichungen zwischen der Theorie und der Wirklichkeit hervor. […] In dieser Beziehung schien mir daher eine Untersuchung der Bewegunsformen, bei denen kein Geschwindigkeitspotential existirt, von Wichtigkeit zu sein.
>
> The influence of friction on liquids could not yet be defined mathematically, yet it is very large in all cases, except for those of infinitely small oscillations, and cause of the greatest discrepancies between theory and reality. […] In this way it seemed to me that an investigation of those forms of motion for which there is no velocity potential would be of importance.

This work introduced the vorticity vector and gave the derivation of its evolution equation, as well as the Biot–Savart law for obtaining the velocity from a given vorticity field in the whole-space. In modern notation these results read respectively:

$$\boldsymbol{\omega} := \nabla \times \mathbf{u}, \tag{1.1}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u}, \tag{1.2}$$

$$\mathbf{u}(\mathbf{x}) = -\frac{1}{4\pi} \int_{\mathbb{R}^3} \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^3} \times \boldsymbol{\omega}(\mathbf{y}) \, \mathrm{d}\mathbf{y}. \tag{1.3}$$

Until Helmholtz, the primary research focus was on potential flows: all known forces of nature were conservative and had a potential of their own; and it can be shown that such forces can only induce potential flows in inviscid fluids. Thus the question arose how a fluid at rest could possibly develop rotational flow patterns. To this end he also introduced vortex lines ('Wirbellinien'), vortex filaments ('Wirbelfäden', literally *vortex threads*), and vortex sheets ('Wirbelflächen', literally *vortex surfaces*). The latter refers to a surface S of discontinuity in the flow field, where the velocity's components normal to the surface are continuous, but the tangential components may jump. He showed that such a discontinuity corresponds to a flow whose vorticity is concentrated on the surface. In modern terminology such a vorticity field is a distribution $\boldsymbol{\omega} \in \mathcal{D}'$ such that:

$$\langle \boldsymbol{\omega}, \boldsymbol{\varphi} \rangle = \int_S \boldsymbol{\omega}_S \cdot \boldsymbol{\varphi} \, \mathrm{d}S \qquad \forall \boldsymbol{\varphi} \in \mathcal{D} = \left(C_0^\infty(\mathbb{R}^3)\right)^3, \tag{1.4}$$

where $\boldsymbol{\omega}_S$ is a smooth function defined on the surface S. For its corresponding velocity field one subsequently obtains:

$$\mathbf{u}(\mathbf{x}) = -\frac{1}{4\pi} \int_S \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^3} \times \boldsymbol{\omega}_S(\mathbf{y}) \, \mathrm{d}S(\mathbf{y}). \tag{1.5}$$

Helmholtz then went on and introduced the famous Löffel-Experiment (from German *Löffel*, 'spoon'). In this thought experiment one stirs a liquid with an idealised, infinitely flat spoon. Due to the stirring motion, the tangential velocity components on both sides of the spoon differ. If one then abruptly pulls the spoon out of the fluid a vortex sheet remains and thus vorticity has entered the flow. Nowadays this experiment is often called Klein's Kaffeelöffel, after a 1910 article of Felix Klein [4], even though he clearly attributes it to Helmholtz. (Klein, however, was the first one to define the liquid as coffee.) Subsequently the study of the motion of vortex sheets was the next major step.

1.1.2 Kelvin–Helmholtz Instability

Eleven years later, in 1869, Helmholtz' observations on the instabilities of vortex sheets were published [5]. He thought about the flow at the exit of an organ pipe, where a jet of air enters the surrounding air at rest, similar to the case depicted in figure 1.1. Similarly, Sir William Thomson—who would later become to be known as Lord Kelvin—discussed the instability of water surfaces subject to winds in 1871 [6].

Here we will briefly discuss Helmholtz' example. Because of the absence of friction, the tangential components of the jet and its surroundings differ, while their normal components are both zero and thus continuous. In other words, a vortex sheet develops between the jet and its surroundings. Using experiments with smoke, Helmholtz observed that the slightest disturbances to such a sheet would cause the flow to become unstable and would result in the sheet to roll up, resulting in the archetypical smoke rings nowadays known from tobacco smokers, as seen in figure 1.2. The question was now whether the previously derived equations for the evolution of the vorticity would also predict such a behaviour. Because there is no closed-form solution of these equations, the answer to this question has proven to be a formidable challenge.

Later Lord Rayleigh published his findings [7] on the linearisation of the two-dimensional version of these equations. The analysis of these linearised equations showed that small disturbances were amplified exponentially over time and thus seemed to confirm the validity of the Euler equations. Subsequently the phenomenon became known as the Kelvin–Helmholtz instability. However, this very same prediction also means that the linearisation quickly ceases to be accurate. Thus the question remained whether the original non-linear equations would correctly predict the vortex sheet's roll-up.

1.1.3 Rosenhead's Vortex Method

It was the question about the evolution of vortex sheets that in 1931 lead Louis Rosenhead to his numerical experiments [9]. To the best of our knowledge, his numerical approach is not only the first manifestation of vortex methods, but also of all particle methods in general.

To simplify matters, Rosenhead considered a vortex line subject to a small sinusoidal disturbance in a two-dimensional, periodic setting. He divided the line into segments of

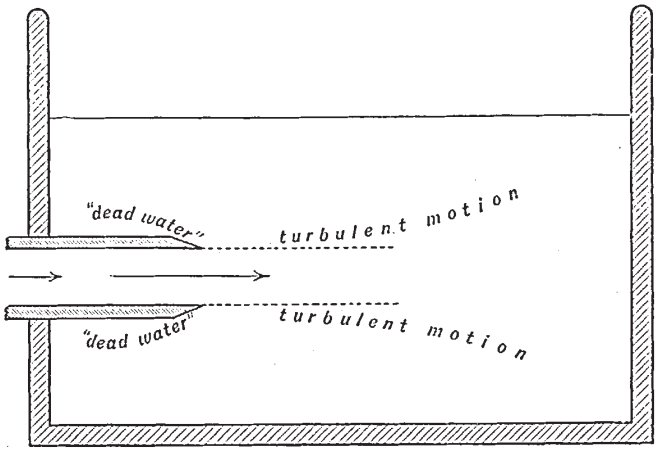

FIG. 2.

Figure 1.1: An illustration of the organ pipe's flow into surroundings at rest. Taken from Kelvin's work on what became to be known as the Kelvin–Helmholtz instability [8]. Copyright expired according to Japanese law, author died more than 50 years ago in 1907.

Figure 1.2: An example of vortex sheet roll-up: smoke rings by a smoker. Copyright © 2008 Andrew Vargas. 'wheres the 2?', https://www.flickr.com/photos/45665234@N00/2891056110. Licensed under the Creative Commons BY license version 2.0: https://creativecommons.org/licenses/by/2.0/legalcode.

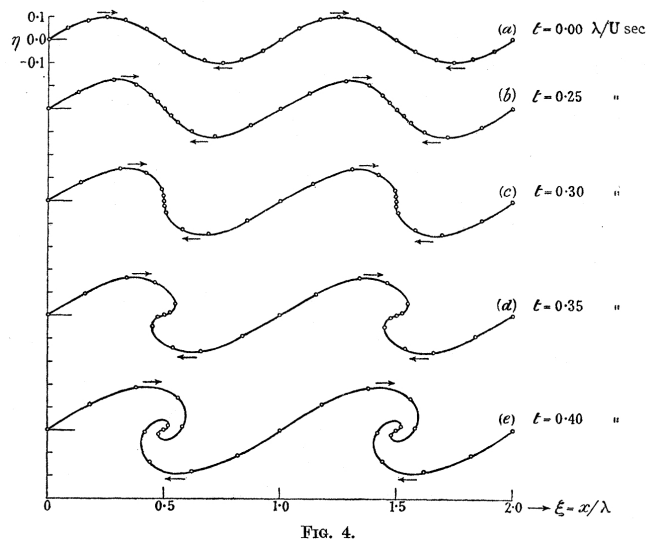

FIG. 4.

Figure 1.3: Rosenhead's results from his vortex particle method [9]. The results suggest that vortex sheets roll up as expected. Copyright © 1931 Louis Rosenhead. The Royal Society, publisher of the original work, allows reproduction of up to five figures from works published more than 70 years ago without need of seeking permission.

equal length h, and approximated these segments by replacing them with point vortices at their centres carrying all of that segments vorticity. This effectively corresponds to approximating the line-integral resulting from the Biot–Savart law with the mid-point quadrature rule. The Biot–Savart law then reduces to a finite sum which can be computed manually. In modern terms this corresponds to an approximation of the vortex line with a sum of weighted Dirac deltas:

$$\omega \approx \omega_h := \sum_{i=1}^{N} \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i), \tag{1.6}$$

where the Γ_i is the amount of vorticity on segment i, and $\mathbf{x}_i$ denotes its centre. His approach was then to move the $\mathbf{x}_i$ according to the resulting velocity field, i. e., modifying the $\mathbf{x}_i$ according to $\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}_i(t), t)$, where $\mathbf{u}$ is obtained from ω_h by use of the Biot–Savart law. For this he used a Runge–Kutta method. Performing his computations by hand, he was limited to fairly small numbers of such *particles*. His computations suggested that vortex sheets do indeed roll up like observed in nature, as depicted in figure 1.3. Later repetitions of Rosenhead's method, however, lead to inconclusive results. Birkhoff and Fisher, for example, reported chaotic particle movement [10]. Subsequently his method received criticism and its convergence was questioned.

There are several reasons for these inconclusive results. For once, due to the instability rounding errors would also be amplified exponentially over time. Illustrations on the

influence of this error were for example later given by Krasny [11, Figures 1 and 3], who compared computations in single and double precision arithmetic. On the other hand, vortex particles are significantly more irregular than vortex lines or sheets: they induce singular velocity fields with velocity values approaching infinity close to particles. In modern mathematical terms this is reflected in the class of test-functions permitted for such functionals: for a vortex sheet the test-function only needs to possess a well-defined trace on the surface, while a vortex point requires its test-function to be continuous in the classic, point-wise sense. This loss of regularity can cause simulations to 'blow up.'

1.1.4 Particle-in-Cell Methods

In the second half of the 1950ies Harlow independently developed the Particle-in-Cell method [12, 13]. In the latter reference the *compressible* Euler equations in their velocity–pressure formulation are considered for a simple, box-shaped domain. The box is subdivided into cubical cells of equal size σ; within each cell velocity, pressure, and energy are assumed to take constant values. Particles representing fluid mass are then placed into each cell. Mathematically this means that the fluid density ρ is discretised using a particle field ρ_h, where h denotes the average distance between neighbouring particles. In order to obtain a density value for a cell, the mass of all particles in that cell is summed up and divided by the cell's volume. This corresponds to a regularisation step, as a piece-wise constant approximation ρ_σ is obtained from a particle field ρ_h, which is only defined as an irregular distribution.

We can thus for the first time differentiate between a particle approximation ρ_h and a regularised approximation ρ_σ on the cells. Although not emphasised in these early works, this regularisation step and its interplay with the other quantities on the cells would prove to be crucial in the modern analysis of particle methods.

Convection of mass is then again modelled by moving the particles, while the other flow equations are solved using a conventional finite difference schemes. This is probably the first hybrid method combining particle methods and conventional mesh-based solvers. In Appendix II for the first time particles are identified as Dirac delta distributions. Subsequent improvements of the particle–grid regularisation scheme lead to the name Cloud-in-Cell method [14].

1.1.5 Further Reading

A very extensive review of the beginnings of vortex dynamics was written by Meleshko and Aref [15]. Slowly but surely, with the development of Prandtl's boundary layer theory [16], the importance of viscous effects in the generation of vorticity became generally accepted. Subsequently, in the later development, the study of vortex sheets took a back seat and most authors studied more regular, viscous flows. For an overview over the later developments of vortex sheet theory we thus refer the interested reader to Chapter 9 of Majda and Bertozzi's book [17]. Much later Krasny's computations using a regularised vortex method clearly confirmed the roll-up of vortex sheets [18].

1.2 Blobs, Cells, and Early Analysis (1973–1985)

The year 1973 saw the publishing of two influential articles which would have a lasting impact on the further development of vortex methods. Christiansen [19] applied the Clound-in-Cell approach to the vorticity equation, resulting in what is now known as the first Vortex-in-Cell method. Chorin [20], on the other hand, introduced blob functions to regularise the particle field as well as the first scheme to handle viscosity and solid boundaries in vortex methods. His blob method allowed for the first rigorous proofs of convergence of his method in certain cases.

1.2.1 Vortex Blob Methods

It is hard to overestimate the impact Chorin's 1973 article [20] had on the development of vortex methods. His method already contains all of the essential ingredients of modern vortex methods. He considered slightly viscous flows past a cylinder in the plane and his approach introduced several novelties at once:

1. Based on physical arguments, Chorin argued that one should consider vortex blobs ζ_σ of core-width σ instead of point vortices. These blobs correspond to smooth approximations of the Dirac delta function. Alternatively, this can be interpreted as particular regularisation of the vortex particle field: $\omega_\sigma = \omega_h \star \zeta_\sigma$, where $\star$ denotes convolution. An approximate velocity field would then be computed according to the Biot–Savart law $\mathbf{u}_\sigma = \mathbf{K} \star \zeta_\sigma \star \omega_h$ and the particles would be convected according to this smooth, approximate velocity.

2. Viscous and inviscid effects would be treated separately. First, one would ignore viscosity, and advance the approximation in time according to the inviscid equations. In a second step viscous effects would be accounted for in the absence of convection. This approach is nowadays called *viscous splitting.*

3. Again based on physical intuition, viscosity was modelled as a Brownian motion of the particles. This was implemented by randomly altering the particle positions according to a specified probability distribution. This approach came to be known as *random walk.*

4. The inviscid, no-through boundary condition on the cylinder was enforced using a boundary element method. The resulting slip on the boundaries could then be cancelled by placing a vortex sheet on the boundaries, which was again approximated using particles.

In numerical experiments his method correctly predicted the typical Kármán vortex street observed in flows past cylinders at certain Reynolds numbers. The success of his method attracted the interest of several researchers and its various new components created a whole set new questions: how should the blob-function ζ_σ look like? How should one choose the blob size σ in relation to the particle spacing h? How large is

the error introduced by the viscous splitting? How accurate is the simulated Brownian motion? And ultimately: what is the method's order of convergence?

These questions could only later be answered by means of rigorous analysis. Until such results became available, different authors tried different choices. Based on physical arguments, for example, Chorin [20] as well as Chorin and Bernard [21] chose $\sigma = h/2\pi$, while Milinazzo and Saffmann [22] chose $\sigma = h/50$.

1.2.2 Early Analysis

Under the term *early analysis* we refer to a class of convergence proofs that heavily rely on the properties of individual particles' trajectories. Convergence proofs of this type were the first to become available for simplified versions of Chorin's method. These proofs have in common that they are very involved and complex.

The first attempt at such an analysis is due to Dushane [23], who studied the inviscid method in the two-dimensional whole space case, i. e., in the absence of boundaries. Hald and Del Petre later spotted an error in his analysis and gave the first proof of convergence for limited times [24]. Hald then later significantly improved this result and showed convergence for all times [25]. His proof showed that the blob function ζ_σ should fulfil moment conditions. These conditions state that for all polynomials p up to a fixed order the blob function ζ_σ should behave exactly like the Dirac delta, i. e., $\int \zeta_\sigma(x)p(x)\,\mathrm{d}x = p(0)$. Interestingly, his result showed that one should choose $\sigma \sim \sqrt{h}$, similar to the result of chapter 4 later in this work.

Beale and Majda gave proofs of convergence for three-dimensional, inviscid vortex methods [26, 27]. Due to the additional vortex stretching term of the three-dimensional equations, their analysis is significantly more complex. They make use of the fact that in unbounded domains the mid-point quadrature rule is of infinite order and conclude that one then can choose $\sigma \sim h$. Numerical experiments and an improvement in the treatment of the stretching term then quickly followed [28, 29]. Hald then later combined all these results, in what may be seen the pinnacle of these early analysis techniques [30].

Greengard and Anderson also analysed the effect of the time discretisation on the method [31]. Their analysis showed that the time-step does not need to fulfil any constraint depending on h or σ to ensure stability; a clear advantage to conventional grid-based methods. The effect of the viscous splitting was analysed by Beale and Majda [32], who showed that the approach converges with first order with respect to the time-step. The convergence of the random walk approach was first proven by Goodman [33]; his result was later improved by Long [34], who showed that the method converges only very slowly. More advanced viscous schemes will be discussed later in chapter 3 of this work.

1.2.3 Vortex-in-Cell Methods

Around the same time as Chorin experimented with vortex blobs, Christiansen applied the Clound-in-Cell approach to the vorticity equation [19]. He considered the inviscid equation in a two-dimensional box, and discretised the vorticity field ω by a set of

particles ω_h. Similar to the Partile-in-Cell approach, a smoothed approximation ω_σ is obtained on the cells by means of a suitable interpolation procedure. Unlike vortex blob methods, the velocity is retrieved from the vorticity by solving the Poisson equation $-\Delta\Psi_\sigma = \omega_\sigma$ for the stream function Ψ_σ using a conventional finite difference scheme. This gives the method an advantage over Chorin's blob method, because such Poisson solvers are significantly faster than the evaluation of the Biot–Savart law. Velocity values at the particle locations were then retrieved by differentiation of the stream function, again using finite differences. He presented the results of several numerical experiments, which seemed to confirm the validity of his approach.

Due to the various sources of error in the method—interpolation, Poisson solver, finite difference differentiation formulae—rigorous analysis of his scheme was difficult and we are unaware of any complete proof of the 'early analysis' type. Later authors applied the method to vortex sheets [35], improved the particle–mesh interpolation scheme [36], and extended it to three-dimensional flows [37].

1.2.4 Further Reading

In 1980 Leonard [38] gave an excellent review of the various vortex methods in use at the time. He also discusses vortex filament methods, a different extension of vortex particle methods to three-dimensional space. In these methods one considers line segments of vorticity. They have some benefits compared to three-dimensional particle methods, but because of the increased cost to track line segments they are less common today. Leonard later also gave a review of these methods [39].

'Smoothed particle hydrodynamics' are particle methods that were developed completely independently of vortex methods in 1977 by Gingold, Monaghan, and Lucy [40, 41]. It is remarkable that these methods also regularise particles by blobs—exactly like Chorin [20]—while coming from a completely different background.

1.3 Modern Analysis and Fast Algorithms (since 1985)

The second half of the 1980ies brought both practical and theoretical advances for vortex methods. On the practical side the development of so-called *fast algorithms* like the treecode [42] and its subsequent improvement, the fast multipole method [43, 44] allowed the handling of huge particle numbers. On the theoretical side the works of Raviart [45] and Cottet [46] significantly simplified analysis and allowed for an entirely new interpretation of vortex methods.

1.3.1 Fast Algorithms

In vortex blob methods the velocity induced by the particle field is evaluated using the Biot–Savart law $\mathbf{u}_\sigma = \mathbf{K} \star \zeta_\sigma \star \omega_h$, where $\mathbf{K}$ denotes the Biot–Savart kernel, ζ_σ the blob function, and ω_h the particle approximation of the vorticity. Setting $\mathbf{K}_\sigma := \mathbf{K} \star \zeta_\sigma$, this yields the representation $\mathbf{u}_\sigma(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{K}_\sigma(\mathbf{x} - \mathbf{x}_i)\Gamma_i$. Subsequently, evaluating this sum at all N particle locations to advance ω_h in time yields a complexity of $\mathcal{O}(N^2)$.

Until the development of the fast algorithms, this prevented the use of large particle numbers and gave vortex-in-cell methods with their mesh-based Poisson solvers a clear advantage.

The treecode [42] and the fast multipole method [43, 44] reduced this complexity to $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$, respectively. The practical implications of these algorithms are dramatic: they would allow computers to handle billions instead of just thousands of particles, as before. For an example of such large scale simulations we refer the reader to the works of the group of Yokota [47].

1.3.2 Modern Analysis

In 1985 Raviart's lecture notes on particle methods were published [45]. Assuming the velocity field would be known, the inviscid vorticity equation forms a linear transport equation. He realised that moving the particles according to the known velocity then corresponds to the analytic solution of the underlying transport equation. The stability of solutions to the transport equation in turn depends on the smoothness of the velocity field, and can be analysed without explicitly looking at particle trajectories. In vortex blob methods, the accuracy and smoothness of the approximate velocity in turn depends on the choice of the blob function ζ_σ.

This lead to the greatly simplified analysis by Cottet [46], who realised that the error of the method can be split into two parts: the first part is the particle error, resulting from the discretisation of the vorticity field. The other part is the error introduced by the regularisation with the blob functions. His approach clearly distinguished between these two sources of error, and also allowed for simplified analysis of the stretching term and of vortex-in-cell schemes.

1.3.3 Further Reading

In 2000 Cottet and Koumoutsakos published their monograph on vortex methods [48]. This work contains many more references, as well as practical and analytical results. A slightly more recent review is given by Koumoutsakos [49].

While the practical implications of the development of fast algorithms were huge, they did not change the underlying structure of vortex methods. For a more detailed treatment of the topic we thus refer the reader to the literature, for example to Pettitt and Kurzak's review [50].

1.3.4 Open Problems

A common feature of all the previously mentioned analysis results is that they assume the whole-space case or periodic boundary conditions. We are unaware of any complete convergence results in the presence of solid boundaries. In fact, all particle methods face severe difficulties at boundaries, as the commonly used blob functions fail to yield accurate regularisations in this case.

Another problem is the unavailability of boundary conditions for the vorticity, especially in the viscous case. The difficulty of boundary conditions and the treatment

of viscosity in vortex methods will be discussed in more detail in the following chapter. As we will see, the viscous term completely changes the character of the solutions to the equations.

After having introduced the necessary mathematical framework, we will see that several of the problems encountered can be considered in isolation. We will describe these problems in greater detail at the end of the next chapter. Our main contributions in this work will be a new scheme for viscous unbounded flows in chapter 3, a truly general solution to the regularisation problem in bounded domains in chapter 4, as well as analytical formulae for the integration of the Biot–Savart integral in chapter 5.

Chapter 2

Vortex Methods

In this chapter we describe vortex methods and their underlying mathematical ideas in greater detail. In order to understand these schemes, we need to introduce the necessary mathematical framework first. We begin by introducing the equations of incompressible flow in both their primitive variable and vorticity–velocity formulations and discuss the necessary boundary conditions. In the following two sections we separately discuss the kinematic and dynamic aspects of the equations. In the inviscid case we will see that the solution to the dynamic part of the equations can be given by tracking fluid particles to their original locations. This motivates the introduction of particle fields as a discretisation method in the following section. We end this chapter by outlining a generic vortex particle method and briefly describe the steps involved. For the mathematical notation used throughout this thesis we refer the reader to appendix A.

2.1 Flow Equations

In this section we briefly introduce the incompressible Navier–Stokes equations in their primitive variable formulation. Their saddle-point structure makes the elimination of the pressure variable desirable, which can be achieved by using the vorticity equation. The resulting set of equations is called the vorticity–velocity formulation, which consists of a dynamic and a kinematic part, which will be discussed in further detail in sections 2.2 and 2.3, respectively. We end this section with a brief discussion of boundary conditions.

2.1.1 Primitive Variable Formulation

Many treatises in the field of computational fluid mechanics start with an introduction of the Navier–Stokes equations, and this one shall be no different. We will restrict ourselves to incompressible flows, which already pose significant challenges, while still accurately describing a wide variety of flows of practical interest. To this end, we assume that we are given a domain $D \subset \mathbb{R}^d$, $d \in \{2, 3\}$, which may but does not need to be bounded. This domain D will be called the *physical domain*. In two- and three-dimensional space and in the absence of any external forces such as gravity, the Navier–Stokes equations read in their simplest form:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\dfrac{1}{\rho}\nabla p + \nu \boldsymbol{\Delta} \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \qquad \text{in } D. \tag{2.1}$$

Here and throughout this work we will use boldface letters to refer to vector quantities, such as the velocity $\mathbf{u}$. The first line is a vector equation describing the conservation of momentum. In there ρ refers to the fluid's density, which by definition is constant in incompressible flows. The scalar field p is the pressure and the parameter $\nu \geq 0$ refers to the fluid's kinematic viscosity, which is also assumed to be constant throughout this work. The symbol $\boldsymbol{\Delta}$ refers to the vector Laplacian. The second line is called the continuity equation and describes the conservation of mass.

For a given domain $D \subset \mathbb{R}^d$, $d \in \{2, 3\}$, and a set of suitable boundary conditions, these seemingly simple equations already form a complete description of incompressible flows with constant viscosity, and it is widely believed that they model complex phenomena such as turbulence and boundary layers correctly. The majority of available flow solvers are based on this formulation in the variables velocity and pressure, which are also called the *primitive variables*.

2.1.2 Vorticity–Velocity Formulation

The continuity equation $\nabla \cdot \mathbf{u} = 0$ does not contain the pressure variable, giving the set of equations (2.1) a saddle-point structure. This means that the pressure is only given implicitly; the continuity equation is a constraint to the momentum equation and the pressure variable acts as a Lagrange multiplier. This causes significant difficulties in their numerical and theoretical treatment. One is therefore interested in eliminating the pressure variable from these equations.

This can be achieved by instead looking at the vorticity. In three-dimensional space the vorticity vector is defined by $\boldsymbol{\omega} := \nabla \times \mathbf{u}$. In two-dimensional space the vorticity reduces to a scalar, which further simplifies matters: $\omega := \partial_{x_1} u_2 - \partial_{x_2} u_1$, where u_1 and u_2 refer to the Cartesian components of the velocity vector. Intuitively, the vorticity describes the tendency of a fluid particle to rotate around its own centre. In three-dimensional space the vorticity vector points in the direction of the rotational axis, in two-dimensional space this is the axis normal to the plane.

Taking the curl of the momentum equation, one then obtains with help of the continuity equation $\nabla \cdot \mathbf{u} = 0$ as well as the vector calculus identities $\nabla \cdot (\nabla \times (\bullet)) \equiv 0$ and $\nabla \times (\nabla(\bullet)) \equiv \mathbf{0}$ the *vorticity equation* in two-dimensional:

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

as well as in three-dimensional space:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} - \boldsymbol{\omega} \cdot \boldsymbol{\nabla}\mathbf{u} = \nu \boldsymbol{\Delta} \boldsymbol{\omega}. \tag{2.3}$$

The term $\boldsymbol{\omega} \cdot \boldsymbol{\nabla}\mathbf{u}$ is called *vorticity stretching* and is of major importance in the understanding of turbulence. The fact that it is missing in the equations for the plane is the main reason why three-dimensional flows are fundamentally more complex than their two-dimensional counterparts. This is further underlined by the fact that in the plane the vorticity is a only a scalar.

Together with the continuity equation and the definition of the vorticity, we obtain the *vorticity–velocity formulation* of the Navier–Stokes equations in two:

$$\begin{cases} \dfrac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega = \nu \Delta \omega, \\ \nabla \cdot \mathbf{u} = 0,\ \partial_{x_1} u_2 - \partial_{x_2} u_1 = \omega, \end{cases} \qquad \text{in } D \subset \mathbb{R}^2, \tag{2.4}$$

and three dimensions:

$$\begin{cases} \dfrac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} - \boldsymbol{\omega} \cdot \boldsymbol{\nabla} \mathbf{u} = \nu \boldsymbol{\Delta} \boldsymbol{\omega}, \\ \nabla \cdot \mathbf{u} = 0,\ \nabla \times \mathbf{u} = \boldsymbol{\omega}, \end{cases} \qquad \text{in } D \subset \mathbb{R}^3. \tag{2.5}$$

In this formulation, the Navier–Stokes equations naturally split into two separate parts. The vorticity equation in the first line describes the dynamics of the system: assuming the velocity field $\mathbf{u}$ is known we are given a convection–diffusion equation for the evolution of the vorticity. The second line describes the system's kinematics: conversely assuming we are given some vorticity field ω or $\boldsymbol{\omega}$, respectively, we are given a closed system for obtaining the corresponding velocity $\mathbf{u}$. There are specialised approaches for the solution of each these problems when considered in isolation; it is their coupling which makes the resulting system so complex. We will further discuss the kinematic and dynamic parts of this system in sections 2.2 and 2.3, respectively.

2.1.3 Conditions on the Domain and the Vorticity

Before we continue our discussions, we will need to impose certain conditions on the physical domain D. We will assume that D is *simply connected*. This for example excludes flows around doughnut shaped objects in three-dimensional space. In two-dimensions the restriction is more severe, as it excludes external flows past any bounded object such as an aerofoil. The reason for this restriction lies in the fact that in simply connected domains *every* curl-free vector field is conservative, i. e., it is the gradient of a scalar potential φ: $\nabla \times \mathbf{u} = \mathbf{0} \iff \mathbf{u} = \nabla \varphi$. In multiply connected domains the implication only holds the other way around. As a consequence the kinematic part of the Navier–Stokes equations would only allow for a unique solution if one prescribes additional boundary conditions. The Kutta condition for two-dimensional flows past an aerofoil is an example of such an additional condition. While in principle vortex methods may be extended to multiply connected domains, it makes sense to restrict our discussion to this simple case, which—as we will see—already poses several challenges.

The following assumptions will allow us to run numerical simulations on computers, which in principle can only handle bounded objects. We will assume that the domain's boundary ∂D has a finite a finite area (or length if $d = 2$). This includes the possibility $\partial D = \emptyset$ and thus also $D = \mathbb{R}^d$, but it excludes cases like the half-space. This condition will allow us to carry out integrations over the domain's boundary in computer simulations.

For the sake of simplicity, we will assume that the vorticity field is compactly supported. This allows us to limit discretisations to bounded domains Ω, such that

$\operatorname{supp}\omega \subset \operatorname{clos}\Omega \subset \operatorname{clos} D$ and $\partial D \subset \partial\Omega$. In practice it will be sufficient for ω to vanish sufficiently fast outside such a domain Ω. This is not a severe restriction, as even in many external flows of practical interest the vorticity *does* essentially vanish outside of a bounded region. The domain Ω will be called the *computational domain*. This restriction will furthermore allow us to avoid various technicalities in the ongoing discussion, for example when measuring errors in the $L^2(\Omega)$-norm. The boundary conditions will still be considered on ∂D only.

2.1.4 Boundary Conditions

In order to solve the Navier–Stokes equations in either of their formulations, we need to prescribe certain boundary conditions at the domain's boundary ∂D. The type of these conditions differs for the inviscid ($\nu = 0$) and viscid case ($\nu > 0$). If we consider external flows such that the physical domain D extends to infinity, we furthermore need to prescribe far-field conditions, corresponding to 'boundary conditions at infinity.' As a concrete example we will consider the flow past a fixed ball $B \subset \mathbb{R}^3$ in space, such that the physical domain is given by $D = \mathbb{R}^3 \setminus B$. Periodic boundary conditions will not be considered in this work, and we refer the interested reader to the book of Majda and Bertozzi [17].

The Far-field Condition

In external flows one needs to prescribe a far-field velocity $\mathbf{u}_\infty$ with zero curl and divergence. In other words, $\mathbf{u}_\infty$ needs to be a potential flow. Given such a velocity field, the far-field condition then reads:

$$|\mathbf{u}(\mathbf{x}) - \mathbf{u}_\infty(\mathbf{x})| = \mathcal{O}(|\mathbf{x}|^{1-d}), \quad |\mathbf{x}| \to \infty. \tag{2.6}$$

In external flows it is common to set $\mathbf{u}_\infty$ to a constant 'free-stream velocity'. In the following discussion we will usually include the far-field velocity $\mathbf{u}_\infty$ in the resulting mathematical expressions. The formulas remain valid for the case of bounded domains by setting $\mathbf{u}_\infty = \mathbf{0}$.

Inviscid Boundary Condition

If one thinks of physical boundary ∂D as the surface of some impermeable body in the flow, one quickly arrives at the no-through boundary condition:

$$\mathbf{u} \cdot \mathbf{n} = \underbrace{\mathbf{u}_{\text{body}} \cdot \mathbf{n}}_{=:g} \qquad \text{on } \partial D, \tag{2.7}$$

where $\mathbf{n}$ denotes the exterior normal vector of the surface ∂D. This condition ensures that the liquid may not enter the surface of the body. In the case of a stationary body like our ball B one would consequently prescribe $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial D = \partial B$. Due to the absence of friction, in inviscid flows ($\nu = 0$) the fluid may slip along the body's surface. We thus do not prescribe the tangential values of the velocity.

In bounded domains D the function g needs to fulfil a compatibility condition. By the continuity equation $\nabla \cdot \mathbf{u} = 0$ one obtains with the help of the divergence theorem:

$$0 = \int_D \nabla \cdot \mathbf{u} \, \mathrm{d}x = \int_{\partial D} \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}S = \int_{\partial D} g \, \mathrm{d}S. \tag{2.8}$$

Viscous Boundary Condition

In viscous flows ($\nu > 0$) the presence of friction prevents the liquid from slipping alongside a body's surface. Here one thus additionally prescribes the tangential components of the velocity at the boundary of the body. In summary one thus prescribes the entire velocity vector:

$$\mathbf{u} = \mathbf{u}_{\text{body}} \qquad \text{on } \partial D. \tag{2.9}$$

In the case of a body at rest as in our example of a flow past a sphere, one would thus prescribe $\mathbf{u} = \mathbf{0}$ on ∂D.

Some Remarks

There also is a mathematical reason for different nature of the viscid and inviscid boundary conditions. In the inviscid case the vorticity equations (2.2) and (2.3) which describe the dynamics of the system are transport equations, i. e., hyperbolic. The viscous equations, on the other hand, contain a diffusion term and are parabolic. The mathematical theory behind hyperbolic and parabolic equations is fundamentally different, resulting in the change of boundary conditions. To avoid the additional complexity that comes with the no-slip condition, in the development of viscous schemes one often first considers the whole-space $D = \mathbb{R}^d$, such that there are no boundaries $\partial D = \emptyset$.

The question if and under which additional conditions these boundary conditions allow for a unique solution of the Navier–Stokes equations is an open problem. In fact, even the question of the mere existence of smooth solutions for the three-dimensional whole-space space case $D = \mathbb{R}^3$ remains unanswered as of the time of writing; it is one of the famous seven Millenium Prize Problems.[1]

2.2 Kinematics

The kinematic part of the Navier–Stokes equations is given by the systems

$$\nabla \cdot \mathbf{u} = 0, \ \partial_{x_1} u_2 - \partial_{x_2} u_1 = \omega \text{ in } D \subset \mathbb{R}^2 \tag{2.10}$$

for the two-dimensional case, and:

$$\nabla \cdot \mathbf{u} = 0, \ \nabla \times \mathbf{u} = \boldsymbol{\omega} \text{ in } D \subset \mathbb{R}^3 \tag{2.11}$$

[1] See `http://www.claymath.org/millennium-problems`.

for the three-dimensional case. In this section we discuss the solution of these systems for a given vorticity. In the three-dimensional case, due to the vector calculus identity $\nabla \cdot (\nabla \times (\bullet)) \equiv 0$, this given vorticity must fulfil $\nabla \cdot \boldsymbol{\omega} = 0$ in D. We are thus looking for a vector field $\mathbf{u}$ with prescribed curl and divergence. This is a classical problem from potential theory.

Note that this problem is independent of the viscosity ν, and thus the same will be the case for its solution. In particular, the solution will be uniquely determined by the far-field condition and the velocity's normal component on the boundary ∂D. The additional tangential velocity components supplied in the viscous case are in fact a condition on the given vorticity field: because the solution to the kinematic equations is uniquely determined by the inviscid boundary condition, the supplied vorticity field must be such that this solution also fulfils the conditions on the velocity's tangential components at the boundary. A vorticity field that fulfils this condition is called *admissible*.

2.2.1 The Biot–Savart Law

We first consider the whole-space case $D = \mathbb{R}^d$ together with the far-field velocity $\mathbf{u}_\infty = \mathbf{0}$. For a given vorticity field, we will denote the solution of this problem by $\mathbf{u}_\omega$ or $\mathbf{u}_{\boldsymbol{\omega}}$, respectively. The unique solution to this problem is given by the Biot–Savart law. For the two-dimensional case we define the kernel function $\mathbf{K}$ as:

$$\mathbf{K}(\mathbf{x}) := -\frac{1}{2\pi|\mathbf{x}|^2}\begin{pmatrix} x_2 \\ -x_1 \end{pmatrix}. \tag{2.12}$$

The Biot–Savart law then reads $\mathbf{u}_\omega = \mathbf{K} \star \omega$, where the symbol $\star$ denotes convolution:

$$\mathbf{u}_\omega(\mathbf{x}) = \int_\Omega \mathbf{K}(\mathbf{x} - \mathbf{y})\omega(\mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{2.13}$$

It is sufficient to only integrate over the bounded computational domain $\Omega \subset D$, because we previously introduced the assumption that ω vanishes outside this region. In the three-dimensional case the Biot–Savart law is given by the matrix kernel:

$$\mathbf{K}(\mathbf{x}) = -\frac{1}{4\pi|\mathbf{x}|^3}\begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}, \tag{2.14}$$

and $\mathbf{u}_{\boldsymbol{\omega}} = \mathbf{K} \star \boldsymbol{\omega}$, respectively.

2.2.2 General Solution

Because the kinematic equations form a linear system, one may add any solution of the homogeneous equations $\nabla \cdot \mathbf{u} = 0, \nabla \times \mathbf{u} = \mathbf{0}$ to $\mathbf{u}_{\boldsymbol{\omega}}$; the resulting function will then still fulfil the original set of equations (2.10) and (2.11). In the whole-space case $D = \mathbb{R}^d$ the general solution of the kinematic equations that fulfils the far-field condition (2.6) is thus given by $\mathbf{u} = \mathbf{u}_\infty + \mathbf{u}_{\boldsymbol{\omega}}$.

Let us finally consider the case that $\partial D \neq \emptyset$ and that we are given inviscid boundary data $\mathbf{u}\cdot\mathbf{n} = g$ on ∂D. We then need to find a solution $\mathbf{u}_\varphi$ of the homogeneous equations, i. e., a potential flow, such that $\mathbf{u}_\varphi \cdot \mathbf{n} = g - (\mathbf{u}_\infty + \mathbf{u}_{\boldsymbol{\omega}}) \cdot \mathbf{n}$ on ∂D. Setting $\mathbf{u}_\varphi := \nabla\varphi$, in the three-dimensional case, for example, this amounts to finding the scalar potential φ that solves:

$$\begin{cases} \Delta\varphi = 0 & \text{in } D, \\ \nabla\varphi \cdot \mathbf{n} = g - (\mathbf{u}_\infty + \mathbf{u}_{\boldsymbol{\omega}}) \cdot \mathbf{n} & \text{on } \partial D, \\ |\varphi(\mathbf{x})| = \mathcal{O}(|\mathbf{x}|^{-1}) & |\mathbf{x}| \to \infty, \\ |\nabla\varphi(\mathbf{x})| = \mathcal{O}(|\mathbf{x}|^{-2}) & |\mathbf{x}| \to \infty. \end{cases} \tag{2.15}$$

This is the classical Neumann problem for the Laplace equation, which is known to be well-posed. In particular, it uniquely determines φ. In the case of a bounded domain D the far-field conditions need to be omitted and φ then is only unique up to a constant. The resulting velocity $\mathbf{u}_\varphi$, however, remains unique. This analogously holds in the two-dimensional case.

The solution of this system can be written in terms of a boundary integral over ∂D. For example, in the three-dimensional case, φ may be written as a double layer potential:

$$\varphi(\mathbf{x}) = (\mathcal{D}\mu)(\mathbf{x}) := \frac{1}{4\pi} \int_{\partial D} \frac{(\mathbf{x}-\mathbf{y})\cdot\mathbf{n}}{|\mathbf{x}-\mathbf{y}|^3} \mu(\mathbf{y}) \,\mathrm{d}S(\mathbf{y}) \qquad \mathbf{x} \in D, \tag{2.16}$$

where the scalar function μ is the searched-for surface density, which usually can only be determined numerically. This can be done with so-called boundary element methods, which only require a mesh of the surface ∂D. A very rigorous and detailed treatment of boundary element methods is given in the book by Sauter and Schwab [51].

A variety of different representations of the same function φ and the resulting velocity $\mathbf{u}_\varphi$ are available. Setting for example $\boldsymbol{\gamma} := \nabla \times (\mu\mathbf{n})$, one may also write the velocity as a boundary integral involving the Biot–Savart kernel $\mathbf{K}$:

$$\mathbf{u}_\varphi(\mathbf{x}) = \int_{\partial D} \mathbf{K}(\mathbf{x}-\mathbf{y}) \boldsymbol{\gamma}(\mathbf{y}) \,\mathrm{d}S(\mathbf{y}) \qquad \mathbf{x} \in D. \tag{2.17}$$

This gives rise for the interpretation of $\boldsymbol{\gamma}$ as a boundary layer which has shrunken to thickness zero, i. e., a *vortex sheet*.

Independent of the particular representation of $\mathbf{u}_\varphi$, the unique general solution to the kinematic equations (2.11) is then given by $\mathbf{u} = \mathbf{u}_\infty + \mathbf{u}_\varphi + \mathbf{u}_{\boldsymbol{\omega}}$, and the analogous expression holds in the two-dimensional case. Because ∂D and Ω are assumed to be bounded, all of these integrals can—at least in principle—be evaluated using a computer.

2.2.3 Explicit Representation for the Viscous Case

Let us now consider the case of viscous boundary conditions, i. e., the entire velocity vector $\mathbf{u}$ is prescribed on the boundary ∂D. As briefly mentioned in the introduction of this section, the additional tangential components are in fact conditions on the given

vorticity field. We therefore consider the case where such an admissible vorticity field is given. In this case the velocity $\mathbf{u}_\varphi$ can be given explicitly in terms of the boundary values. This can be achieved with the help of the Helmholtz decomposition. For reasons of brevity, we will again only consider the three-dimensional case.

PROPOSITION 2.1 (HELMHOLTZ DECOMPOSITION). *Let $D \subset \mathbb{R}^3$ be a domain in three-dimensional space and let $\mathbf{u}_\infty$ denote a potential flow. Then **any** sufficiently smooth vector field $\mathbf{u} : D \to \mathbb{R}^3$ that satisfies the far-field condition (2.6) can be written in terms of a scalar potential φ and a vector stream function $\boldsymbol{\Psi}$ as:*

$$\mathbf{u} = \mathbf{u}_\infty - \nabla\varphi + \nabla \times \boldsymbol{\Psi}, \tag{2.18}$$

where φ and $\boldsymbol{\Psi}$ are given by:

$$\varphi(\mathbf{x}) = \frac{1}{4\pi}\int_D \frac{\nabla_{\mathbf{y}} \cdot \mathbf{u}(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,\mathrm{d}V(\mathbf{y}) \;-\; \frac{1}{4\pi}\int_{\partial D} \frac{\mathbf{n}\cdot(\mathbf{u}-\mathbf{u}_\infty)(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,\mathrm{d}S(\mathbf{y}), \tag{2.19}$$

$$\boldsymbol{\Psi}(\mathbf{x}) = \frac{1}{4\pi}\int_D \frac{\nabla_{\mathbf{y}} \times \mathbf{u}(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,\mathrm{d}V(\mathbf{y}) - \frac{1}{4\pi}\int_{\partial D} \frac{\mathbf{n}\times(\mathbf{u}-\mathbf{u}_\infty)(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\,\mathrm{d}S(\mathbf{y}), \tag{2.20}$$

where $\mathbf{n}$ denotes the surface normal exterior to D and the symbol $\nabla_{\mathbf{y}}$ signifies that the derivative shall be taken with respect to the $\mathbf{y}$ variable. In the case $D = \mathbb{R}^3$ the boundary integrals vanish. In bounded domains one has $\mathbf{u}_\infty = \mathbf{0}$.

Remark 2.2. There is a is a bewildering amount of different versions of this statement, a recent review of more than a hundred articles is given by Bhatia et al [52]. Sauter and Schwab give a rigorous treatment of the boundary integrals involved and many references in the first three chapters of their book [51]. A sketch of the proof in the context of vortex methods is also given by Cottet and Koumoutsakos [48, pp. 281f].

Substituting (2.11) into (2.18) to (2.20), one obtains an explicit representation formula for the velocity $\mathbf{u}$, which naturally splits into far-field, boundary, and volumetric terms:

$$\mathbf{u} = \mathbf{u}_\infty + \mathbf{u}_{\partial D} + \mathbf{u}_{\boldsymbol{\omega}}, \tag{2.21}$$

where $\mathbf{u}_{\boldsymbol{\omega}}$ is the velocity from Biot–Savart law from subsection 2.2.1 and the boundary term $\mathbf{u}_{\partial D}$ is given by

$$\mathbf{u}_{\partial D}(\mathbf{x}) := -\frac{1}{4\pi}\int_{\partial D} \frac{(\mathbf{x}-\mathbf{y})\big[(\mathbf{u}-\mathbf{u}_\infty)\cdot\mathbf{n}\big] + (\mathbf{x}-\mathbf{y})\times\big[(\mathbf{u}-\mathbf{u}_\infty)\times\mathbf{n}\big]}{|\mathbf{x}-\mathbf{y}|^3}\,\mathrm{d}S(\mathbf{y}). \tag{2.22}$$

This expression only depends on the boundary values of $\mathbf{u}$. Because the solution of the kinematic equations (2.11) is unique, by comparing the terms we obtain that $\mathbf{u}_{\partial D} = \mathbf{u}_\varphi$.

Let us now briefly consider the case that the entire velocity vector is prescribed, but that the vorticity field is inadmissible. In this case the problem (2.11) *has no solution.* However, the velocities $\mathbf{u}_{\partial D}$ and $\mathbf{u}_\varphi$ are still well defined, and in this case we have $\mathbf{u}_{\partial D} \neq \mathbf{u}_\varphi$. The difference between the two may then be seen as some sort of measure of the inadmissibility of $\boldsymbol{\omega}$.

2.2.4 Properties of the Biot–Savart Integral

In a numerical scheme that uses the vorticity as its primary unknown, one can only assume the availability of an approximation $\boldsymbol{\omega}_\sigma$ of the exact vorticity $\boldsymbol{\omega}$, where σ refers to a discretisation parameter. In general, this approximate vorticity field cannot be assumed to be admissible. Thus, the question arises what is known about the corresponding approximate velocity $\mathbf{K} \star \boldsymbol{\omega}_\sigma$.

The first question we will address is that of the existence and boundedness of the Biot–Savart integral. With help of the Calderón–Zygmund inequality [53, 54] one can show that for all $\boldsymbol{\omega} \in \left(L^p(\Omega)\right)^3$, $p \in (1, \infty)$, one has:

$$\|\mathbf{K} \star \boldsymbol{\omega}\|_{W^{1,p}(\Omega)} \leq C \|\boldsymbol{\omega}\|_{L^p(\Omega)}. \tag{2.23}$$

Here and throughout this thesis the symbol C will refer to a generic positive constant that is independent of the functions involved. This inequality analogously holds for the two-dimensional case. In a nutshell, it tells us that the velocity field $\mathbf{u}_{\boldsymbol{\omega}}$ and its first derivatives continuously depend on the given vorticity, and that this vorticity only needs to be an $L^p(\Omega)$-function. In particular, this inequality also holds true for vorticity fields that are not admissible or do not fulfil $\nabla \cdot \boldsymbol{\omega} = 0$. Now, noting that the Biot–Savart law is linear in the vorticity, we obtain:

$$\|\mathbf{K} \star \boldsymbol{\omega} - \mathbf{K} \star \boldsymbol{\omega}_\sigma\|_{W^{1,p}(\Omega)} = \|\mathbf{K} \star (\boldsymbol{\omega} - \boldsymbol{\omega}_\sigma)\|_{W^{1,p}(\Omega)} \leq C \|\boldsymbol{\omega} - \boldsymbol{\omega}_\sigma\|_{L^p(\Omega)}. \tag{2.24}$$

This means that a good vorticity approximation will also lead to a good velocity approximation, even if it is not admissible. Also note that $\mathbf{K} \star \boldsymbol{\omega} = \nabla \times \mathbf{A}$, with $\mathbf{A}$ corresponding the volume integral in equation (2.20). By the vector calculus identity $\nabla \cdot (\nabla \times (\bullet)) \equiv 0$, we obtain that velocity fields $\mathbf{u}_{\boldsymbol{\omega}}$ induced by the Biot–Savart law *always* fulfil the continuity equation $\nabla \cdot \mathbf{u}_{\boldsymbol{\omega}} = 0$. We cannot, however, assume any more that one also has $\nabla \times (\mathbf{K} \star \boldsymbol{\omega}_\sigma) = \boldsymbol{\omega}_\sigma$.

2.3 Dynamics

In the previous section we assumed we were given a prescribed vorticity and showed how to obtain the corresponding velocity $\mathbf{u}$. In this section we will consider the opposite case: at any time we are given a velocity field $\mathbf{u}$ that satisfies $\nabla \cdot \mathbf{u} = 0$ and ask for the evolution of the vorticity. It is determined by the solution of the vorticity equations (2.2) and (2.3) for the two- and three-dimensional cases, respectively. As mentioned before, the theory for the viscous and inviscid versions of these equations differs significantly, and will thus be discussed in separate sections.

2.3.1 Inviscid Case

In this section we will briefly discuss equations of the form:

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega - b\omega = f \quad \text{in } D, \tag{2.25}$$

where $\mathbf{u}$ is a sufficiently smooth velocity field that satisfies $\nabla \cdot \mathbf{u} = 0$ and $\mathbf{u} \cdot \mathbf{n} = 0$ on ∂D, and b and f are sufficiently smooth functions. The two-dimensional, inviscid vorticity equation (2.2) is of this type with $b = f = 0$. The three-dimensional equation (2.3) corresponds to a system version of (2.25), with a matrix function $\mathbf{B} = (\boldsymbol{\nabla}\mathbf{u})^\top$ and right-hand side vector $\mathbf{f} = \mathbf{0}$. Such equations describe transport phenomena. For this reason in the inviscid case equation (2.2) is also called the vorticity transport equation.

For the solution of this equation let us consider some imaginary 'particle' at some point $\mathbf{x} \in D$ at some time t. Here, by particle we mean an infinitesimal flow volume. We are interested in how this particle changes its position over time as it follows the flow $\mathbf{u}$. The curve that this particle moves along is called its *trajectory*, in the mathematical literature this curve is called a *characteristic*. We will write $\mathbf{X}(\tau; \mathbf{x}, t)$ for the trajectory of the particle that was at position $\mathbf{x}$ at time t. The parameter τ will then give us the position of that particular particle at any other time τ. With these definitions we obtain the following ordinary differential equation for the trajectory of that particle:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\tau}(\tau; \mathbf{x}, t) &= \mathbf{u}(\mathbf{X}(\tau; \mathbf{x}, t), \tau), \\ \mathbf{X}(t; \mathbf{x}, t) &= \mathbf{x}. \end{aligned} \tag{2.26}$$

For Lipschitz continuous $\mathbf{u}$, this system always has a unique solution, and thus $\mathbf{X}$ is uniquely determined. The unique analytic solution of the transport equation is then given by [48, Theorem A.2.1]:

$$\begin{aligned} \omega(\mathbf{x}, t) = \omega\big(\mathbf{X}(0; \mathbf{x}, t), 0\big) \exp\!\left(\int_0^t b\big(\mathbf{X}(\tau; \mathbf{x}, t), \tau\big)\, \mathrm{d}\tau\right) \\ + \int_0^t f\big(\mathbf{X}(\tau; \mathbf{x}, t), \tau\big) \exp\!\left(\int_\tau^t b\big(\mathbf{X}(s; \mathbf{x}, t), s\big)\, \mathrm{d}s\right) \mathrm{d}\tau. \end{aligned} \tag{2.27}$$

The solution of the system in three dimensions can be derived in a similar way. Thus, at any time $t > 0$, the solution of equation (2.25) at position $\mathbf{x} \in D$ can be obtained by tracking that particle along its trajectory to its initial position at $t = 0$. Because we assumed the no-through condition on ∂D, this position is guaranteed to be within the interior of D. Because $\nabla \cdot \mathbf{u} = 0$, no two particles ever collide, such that this 'backtracking' can always be done. As a consequence, *the solution of the transport equation requires only initial values and no boundary conditions.* This confirms that the inviscid boundary conditions described in subsection 2.1.4 are in deed of the correct type. The observation that the solution depends on the trajectory of imaginary particles is crucial to the idea of particle methods, in which this tracking of particles is done in an explicit manner.

2.3.2 Viscous Case

In the case of viscous flows ($\nu > 0$) the vorticity equations (2.2) and (2.3) are (a system of) second order differential equations. Unlike in the inviscid case, the solution of these

equations can generally only be determined numerically. We thus focus on a special case in order to highlight the differences between the viscous and inviscid equations.

The mathematical nature of convection diffusion equations is determined by the highest order spatial derivative, i. e., the viscous term $\nu\Delta\omega$. Let us thus consider the case $D = \mathbb{R}^d$ and $\mathbf{u} \equiv 0$. In this special case, we obtain the so-called heat equation in the whole space:

$$\frac{\partial \omega}{\partial t} = \nu\Delta\omega \text{ in } \mathbb{R}^2, \tag{2.28}$$

or the equivalent vector version for $\boldsymbol{\omega}$ in the case $d = 3$. For this particular case an analytic solution can be given. Defining the heat kernel H:

$$H(\mathbf{x}, \tau) := \frac{1}{(4\pi\tau)^{d/2}} \exp\left(-\frac{|\mathbf{x}|^2}{4\tau}\right), \tag{2.29}$$

the solution of (2.28) is given by $\omega(\mathbf{x}, t) = H(\cdot, \nu t) \star \omega(\cdot, 0)$:

$$\omega(\mathbf{x}, t) = \int_{\mathbb{R}^d} H(\mathbf{x} - \mathbf{y}, \nu t)\omega(\mathbf{y}, 0)\,\mathrm{d}\mathbf{y}, \qquad t > 0. \tag{2.30}$$

Already here one can see major differences when compared to the inviscid case. The solution is now given by a convolution with a kernel function, similar to the Biot–Savart law in subsection 2.2.1. Especially note that the solution at some fixed point in space depends on the initial data at *every point* in space. It is thus no longer sufficient to 'track' imaginary particles back to their origins. One also says that in parabolic equations the information spreads at an infinite speed: an initially compactly supported ω will *immediately* evolve into a function with global support, though exponentially decaying as $|\mathbf{x}| \to \infty$.

If the domain D has boundaries, one will need to prescribe boundary conditions for ω on ∂D. One can for example either prescribe ω itself, i. e., Dirichlet data, or its normal derivative $\nabla\omega \cdot \mathbf{n}$, i. e., Neumann data. In Robin type boundary conditions one prescribes a mix of the two. This remains true for the general case when $\mathbf{u} \neq 0$: then we still are given a parabolic equation, and the unique solution of such equations always requires boundary conditions.

This is problematic in flow problems, where vorticity boundary values are usually unavailable and furthermore vary with time. As described earlier in section 2.2, the prescribed tangential components of the velocity vector at the boundary ∂D are essentially a condition on the vorticity field to remain admissible. One is thus faced with the task of finding the right boundary condition on ω, such that the vorticity field remains admissible. One may try to find the corresponding Dirichlet conditions on ω, but the more common approach is to find the Neumann data. The reason for this is that finding the Neumann data is equivalent to determining the vorticity generation at the wall, and thus allows for a natural interpretation. In an initially irrotational flow field the boundaries are *the only source of vorticity*. One way to obtain this Neumann data is to look at the difference of the velocities $\mathbf{u}_{\partial D}$ and $\mathbf{u}_\varphi$ as introduced in subsection 2.2.3. A detailed explanation of this process can be found in section 6.3.3 of Cottet and Koumoutsakos' book [48].

2.3.3 Some Remarks on Turbulent Flows

Many flows of practical interest are turbulent. Such flows can be characterised by their high Reynolds number, which corresponds to velocity values that are large compared to the kinematic viscosity: $|\mathbf{u}| \gg \nu > 0$. In this case the vorticity equations remain strictly parabolic, but in large areas of the domain its solutions behave like solutions to the inviscid, hyperbolic equations.

At the boundaries, however, inviscid flows may slip, while viscid ones may not. For this reason one observes boundary layers in turbulent flows, which ensure that the no-slip condition can be fulfilled. With shrinking viscosity the thickness of the boundary layers decreases, eventually collapsing to vortex sheets γ or $\boldsymbol{\gamma}$ in the inviscid limit, corresponding to a potential flow like $\mathbf{u}_\varphi$ in section 2.2. On the other hand, the no-slip condition is the only source of vorticity: without the no-slip condition an initially irrotational flow will remain so for all times.

One thus faces a dilemma: in large portions of the domain the flow behaves like an inviscid one, for which particle schemes are a natural approach. On the other hand viscosity is required in order to understand turbulence and boundary layers. Viscous effects can well be handled with mesh based methods and it is not immediately obvious how this can be achieved with particle schemes. Mesh based methods on the other hand typically face severe problems when dealing with transport phenomena. The questions that arise here are: how can viscous effects be efficiently modelled in particle methods? How can particle methods be coupled with a mesh based method? These are questions of ongoing research and we hope that this work forms a small contribution in this direction. For further information on mesh based methods and parabolic equations in general we refer the interested reader to the book of Thomée [55].

2.4 Quadrature Rules and Particle Fields

In the previous sections we have discussed mathematical properties of the flow equations. We have seen how the inviscid equations can be solved by tracking imaginary particles and how the velocity can be computed from the vorticity with help of the Biot–Savart integral. It is thus natural to ask how such particles can be modelled in a computer and how integrals can be approximated numerically. In this section we introduce quadrature rules and we will see that they allow for a natural interpretation as particle fields.

A quadrature rule for a computational domain $\Omega \subset \mathbb{R}^d$ is a set of finitely many *nodes* $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, N$, with associated *weights* $w_i \in \mathbb{R}$, such that for sufficiently smooth functions f it holds that:

$$\int_\Omega f(\mathbf{x}) \, \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{N} w_i f(\mathbf{x}_i). \tag{2.31}$$

For $m \in \mathbb{N}_0$, let $\mathbb{P}_m$ denote the space of polynomials on $\mathbb{R}^d$ of total degree not greater than m. One says a quadrature rule has exactness degree m, if it is exact for all

$f \in \mathbb{P}_m$:

$$\int_\Omega f(\mathbf{x}) \,\mathrm{d}\mathbf{x} = \sum_{i=1}^{N} w_i f(\mathbf{x}_i) \qquad \forall f \in \mathbb{P}_m. \tag{2.32}$$

In the remainder of this section we will discuss how such quadrature rules can be constructed, their accuracy, and their relation to particle approximations.

2.4.1 Meshes

In general, finding good quadrature rules for arbitrary domains $\Omega \subset \mathbb{R}^d$ is a hard problem. Instead one typically considers a very limited set of reference domains $\hat{T}$, which are then mapped to physical domains $T = \Phi(\hat{T})$ by help of mappings Φ and the transformation theorem:

$$\int_{\Phi(\hat{T})} f \,\mathrm{d}\mathbf{x} = \int_{\hat{T}} (f \circ \Phi) |\det \Phi'| \,\mathrm{d}\hat{\mathbf{x}}. \tag{2.33}$$

For affine maps Φ, the resulting quadrature rules for T have the same exactness degree as the quadrature rule for the reference domain $\hat{T}$. The classical and well established example of quadrature rules for the reference interval $\hat{T} = [-1, 1]$ are the Gauß–Legendre formulae [56], which have positive weights $w_i > 0$ and the optimal exactness degree $m = 2N - 1$. For triangles and tetrahedra the theory is less complete. Efficient quadrature rules with positive weights have for example been published by Zhang, Cui, and Liu [57].

A very common strategy encountered in practice is to subdivide the domain Ω into several disjoint simplices, i. e., into intervals, triangles, or tetrahedra, for $d = 1, 2, 3$, respectively. This process is commonly called *meshing*. The resulting subdivision $\mathcal{G} = \{T_1, T_2, \ldots\}$ is called a *mesh*, *grid*, or—in the case that all of the T_i are simplices—a *triangulation*. The elements of a mesh $\mathcal{G}$ are also referred to as *cells*. Simplices have the advantage that they can *always* be transformed to a reference simplex $\hat{T}$ with the help of affine maps Φ.

Exactly decomposing a domain into simplices is of course only possible if it is a polytope, e. g., a polyhedron in three-dimensional space. There are approaches to use curved, so-called 'iso-parametric' cells, which further complicate the analysis. But even with the help of such cells, meshing and integrating over arbitrary domains remains a hard problem for which there is no panacea. In practice one will thus usually need to be satisfied with, e. g., polyhedral approximations of the domain of interest. For such polyhedral domains there are powerful mesh generators available, for example Gmsh [58] or TetGen [59].

In the rest of this thesis we will only consider shape-regular, quasi-uniform triangulations. For a precise definition of these terms we refer the reader to Brenner and Scott's book on finite elements [60, Definition (4.4.13)]; in essence they mean that all cells of a triangulation are of approximately the same size and have a comparable diameter, which is commonly labelled h. We will write $\mathcal{G}_h$ for a mesh of a domain Ω whose cells are of size h. These conditions also ensure that the cells remain reasonably well-shaped,

e. g., in two-dimensional space, triangles have a height comparable to their diameter and do not collapse to line segments as $h \to 0$.

2.4.2 Interpretation as Functionals

For a given domain Ω, the integral can be interpreted as a linear operator I, that maps an integrable function f to a real number $I(f)$:

$$I : L^1(\Omega) \to \mathbb{R}, \qquad f \mapsto \int_\Omega f \,\mathrm{d}\mathbf{x}. \tag{2.34}$$

A linear operator that maps an element of a vector space to a scalar is commonly called a *functional*. Like the integral operator itself, a quadrature rule of exactness degree m can be interpreted as a functional I_m:

$$I_m : C(\Omega) \to \mathbb{R}, \qquad f \mapsto \sum_{i=1}^{N} w_i f(\mathbf{x}_i). \tag{2.35}$$

Note that this functional requires its argument to be at least a continuous function. This is necessary, because otherwise the function evaluation at the quadrature nodes $\mathbf{x}_i$ would not be well-posed. The integral operator I, on the other hand, requires its argument only to be integrable; it is, for example, also well-posed for functions that contain jumps.

The so-called Dirac functional δ evaluates a function at the origin:

$$\delta : C(\Omega) \to \mathbb{R}, \qquad f \mapsto f(\mathbf{0}), \tag{2.36}$$

that is $\delta(f) := f(\mathbf{0})$. For notational convenience we will write $\delta(\mathbf{x} - \mathbf{x}_i)$ for the functional that evaluates a given function f at $\mathbf{x}_i$, i. e., $\delta(\mathbf{x} - \mathbf{x}_i)(f) := f(\mathbf{x}_i)$. With the help of this definition, quadrature rules may equivalently be written as:

$$I_m = \sum_{i=1}^{N} w_i \delta(\mathbf{x} - \mathbf{x}_i). \tag{2.37}$$

2.4.3 Error Bounds

For quadrature rules with positive weights and exactness degree m, classical error bounds are typically of the form:

$$\left| \int_T f(\mathbf{x}) \,\mathrm{d}\mathbf{x} - \sum_{i=1}^{N} w_i f(\mathbf{x}_i) \right| \leq C h^{m+1} \max_{|\alpha|=m+1} \max_{\boldsymbol{\xi} \in T} |\partial^\alpha f(\boldsymbol{\xi})|, \tag{2.38}$$

where $h := \operatorname{diam} T$ denotes the diameter of the cell T and α is a multi-index. We can already see here that high-order approximations typically require equally high smoothness of the function f.

The problem with these estimates is that they do not fit well into the framework of Sobolev spaces, as they require the function f to be strongly differentiable. We

therefore now consider functions $f \in W^{m+1,1}(\Omega)$, $m \in \mathbb{N}_0$. In order for the application of quadrature rules to be well defined, we will additionally need to require that $f \in C(\Omega)$, i. e., that f is at least continuous. Due to the Sobolev embedding theorem, this is automatically fulfilled for any $f \in W^{m+1,1}(\Omega)$ whenever $m + 1 \geq d$. One has the following result:

THEOREM 2.3. *Let $T \in \mathcal{G}_h$ be a cell from a shape-regular, quasi-uniform triangulation of the domain Ω of mesh width h. Let I_m denote a quadrature rule of exactness degree m for the cell T with positive weights. One then has for $m + 1 \geq d$:*

$$\left| \int_T f \, \mathrm{d}\mathbf{x} - I_m(f) \right| \leq C h^{m+1} |f|_{W^{m+1,1}(T)}. \tag{2.39}$$

Proof. Because of the Bramble–Hilbert lemma [60, Lemma (4.3.8)] there exists a polynomial $\mathcal{P}f$ of total degree less than $m + 1$ such that:

$$|f - \mathcal{P}f|_{W^{l,1}(T)} \leq C h^{m+1-l} |f|_{W^{m+1,1}(T)} \qquad \forall l \in \mathbb{N}_0,\ l \leq m + 1. \tag{2.40}$$

Noting that I_m integrates $\mathcal{P}f$ exactly one obtains:

$$\left| \int_T f \, \mathrm{d}\mathbf{x} - I_m(f) \right| = \left| \int_T (f - \mathcal{P}f) \, \mathrm{d}\mathbf{x} - I_m(f - \mathcal{P}f) \right|. \tag{2.41}$$

By the properties of $\mathcal{P}f$ one immediately obtains $|\int_T (f - \mathcal{P}f) \, \mathrm{d}\mathbf{x}| \leq C h^{m+1} |f|_{W^{m+1,1}(T)}$. For the other part one has with Hölder's inequality, the Sobolev embedding, and again the properties of $\mathcal{P}f$:

$$\begin{aligned} |I_m(f - \mathcal{P}f)| = |\sum_{i=1}^{N} w_i (f - \mathcal{P}f)(\mathbf{x}_i)| &\leq \underbrace{\left(\sum_{i=1}^{N} w_i \right)}_{= |T| \sim h^d} \max_i |(f - \mathcal{P}f)(\mathbf{x}_i)| \\ &\leq C h^d \|f - \mathcal{P}f\|_{C(T)} \leq C h^d \|f - \mathcal{P}f\|_{W^{d,1}(T)} \leq C h^{m+1} |f|_{W^{m+1,1}(T)}. \end{aligned} \tag{2.42}$$

□

By summing over the individual cells T one obtains an analogous bound for the composite quadrature rule of the entire domain Ω.

2.4.4 Particle Fields

A particle field u_h is a quadrature rule for integrating sufficiently smooth functions φ against an underlying, sufficiently smooth function u; that is:

$$u_h = \sum_{i=1}^{N} U_i \delta(\mathbf{x} - \mathbf{x}_i), \tag{2.43}$$

such that:

$$\int_\Omega u \varphi \, \mathrm{d}\mathbf{x} \approx u_h(\varphi) = \sum_{i=1}^{N} U_i \varphi(\mathbf{x}_i). \tag{2.44}$$

The reason for the name particle field will become clear in the next subsection, where we discuss the solution of the vorticity equations for a given velocity field and a particle field approximation ω_h of the vorticity ω. When seen as particle fields, quadrature rules I_m may be interpreted as an approximation of the constant function $u = 1$. One can thus see that the weights U_i *are not function values.*

We need to note here that the Biot–Savart kernels $\mathbf{K}$ from equations (2.12) and (2.14) are not smooth functions. Applying a particle approximation for an underlying vorticity field to the Biot–Savart kernel will result in a function that is singular at the particle locations $\mathbf{x}_i$.

Particle fields can be constructed with help of quadrature rules I_m as shown below. For the approach based on radial basis functions, we refer the reader to the work of Barba [61].

THEOREM 2.4. *Let $p \in [1, \infty]$ and let q denote the index conjugate to p, such that $1 = 1/p + 1/q$. Let $m + 1 > \max\{d/p, d/q\}$ or $m + 1 \geq d$ if $p = 1$ or $p = \infty$. Let $T \in \mathcal{G}_h$ denote a cell from a quasi-uniform, shape-regular triangulation of the domain Ω and let I_m denote a quadrature rule of exactness degree m with positive weights for the cell T. For a given function $u \in W^{m+1,q}(T)$ let $u_h := \sum_{i=1}^{N} w_i u(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)$ denote the particle field approximation defined by the weights w_i and nodes $\mathbf{x}_i$ of I_m. One then has for $h > 0$ small enough the following error bound:*

$$\|u - u_h\|_{W^{-(m+1),q}(T)} \leq Ch^{m+1}\|u\|_{W^{m+1,q}(T)}. \tag{2.45}$$

Proof. The proof is similar to that of theorem 2.3. Let $\varphi \in W^{m+1,p}(T)$ be arbitrary but fixed. Because of the Sobolev embedding theorem both u and φ are continuous, such that the particle field $u_h \in W^{-(m+1),q}(T)$ is well defined. Furthermore one obtains by Hölder's inequality that $u\varphi \in W^{m+1,1}(T)$. By the Bramble–Hilbert lemma there thus exists a polynomial $\mathcal{P}u\varphi$ of total degree less than $m + 1$ such that:

$$|u\varphi - \mathcal{P}u\varphi|_{W^{l,1}(T)} \leq Ch^{m+1-l}|u\varphi|_{W^{m+1,1}(T)} \qquad \forall l \in \mathbb{N}_0,\ l \leq m + 1. \tag{2.46}$$

Because I_m integrates $\mathcal{P}u\varphi$ exactly one has:

$$\left|\int_T u\varphi \,\mathrm{d}\mathbf{x} - I_m(u\varphi)\right| = \left|\int_T (u\varphi - \mathcal{P}u\varphi)\,\mathrm{d}\mathbf{x} - I_m(u\varphi - \mathcal{P}u\varphi)\right|. \tag{2.47}$$

For the first term we obtain with the properties of $\mathcal{P}u\varphi$ and Hölder's inequality: $|\int_T (u\varphi - \mathcal{P}u\varphi)\,\mathrm{d}\mathbf{x}| \leq Ch^{m+1}|u\varphi|_{W^{m+1,1}(T)} \leq Ch^{m+1}\|u\|_{W^{m+1,q}(T)}\|\varphi\|_{W^{m+1,p}(T)}$. Remember that $\mathcal{P}u\varphi$ is a polynomial and thus a smooth function. Furthermore, because both u and φ are continuous so is their product and we may write again using Hölder's inequality:

$$|I_m(u\varphi - \mathcal{P}u\varphi)| \leq \underbrace{\left(\sum_{i=1}^{N} w_i\right)}_{=|T|\sim h^d} \|u\varphi - \mathcal{P}u\varphi\|_{C(T)} \leq |T|\|u\varphi - \mathcal{P}u\varphi\|_{C(T)}. \tag{2.48}$$

Let us first consider the case $m+1 \geq d$. In this case we can apply Sobolev embedding theorem to the product $u\varphi$ as in the proof of theorem 2.3 and obtain:

$$|T|\|u\varphi - \mathcal{P}u\varphi\|_{C(T)} \leq Ch^d \|u\varphi - \mathcal{P}u\varphi\|_{W^{d,1}(T)} \leq Ch^{m+1}\|u\|_{W^{m+1,q}(T)}\|\varphi\|_{W^{m+1,p}(T)} \tag{2.49}$$

which yields the desired result. Let us now consider the case that $m+1 < d$ such that we cannot apply the Sobolev embedding. In this case we obtain with help of the triangle inequality and the trivial estimate $\|u\varphi\|_{C(T)} \leq \|u\|_{C(T)}\|\varphi\|_{C(T)}$:

$$Ch^d\|u\varphi - \mathcal{P}u\varphi\|_{C(T)} \leq Ch^d(\|u\|_{C(T)}\|\varphi\|_{C(T)} + \|\mathcal{P}u\varphi\|_{C(T)}). \tag{2.50}$$

Now we can apply the Sobolev embedding to $\|u\|_{C(T)}$ and $\|\varphi\|_{C(T)}$ separately. For the remaining term note again that $\mathcal{P}u\varphi$ is a polynomial on T, so we can make use of the following inverse estimate [60, Lemma (4.5.3)]:

$$\|\mathcal{P}u\varphi\|_{C(T)} = \|\mathcal{P}u\varphi\|_{L^\infty(T)} \leq Ch^{-d}\|\mathcal{P}u\varphi\|_{L^1(T)}. \tag{2.51}$$

With this we obtain with the properties of $\mathcal{P}u\varphi$:

$$\begin{aligned}\|\mathcal{P}u\varphi\|_{L^1(T)} &\leq \|u\varphi - \mathcal{P}u\varphi\|_{L^1(T)} + \|u\varphi\|_{L^1(T)} \\ &\leq Ch^{m+1}\|u\|_{W^{m+1,q}(T)}\|\varphi\|_{W^{m+1,p}(T)} + \|u\varphi\|_{L^1(T)}. \end{aligned} \tag{2.52}$$

It remains to show that $\|u\varphi\|_{L^1(T)} \leq Ch^{m+1}\|u\|_{W^{m+1,q}(T)}\|\varphi\|_{W^{m+1,p}(T)}$. For this note that the cell T is bounded so that we can make use of the continuous embedding $L^\infty(T) \subset L^1(T)$. Thus:

$$\|u\varphi\|_{L^1(T)} \leq |T|\|u\varphi\|_{L^\infty(T)} \leq Ch^d\|u\|_{C(T)}\|\varphi\|_{C(T)}. \tag{2.53}$$

Again applying the Sobolev embedding to u and φ separately yields the result. □

By summing over the all the cells T of the mesh $\mathcal{G}_h$, one obtains an analogous estimate for the error bound of the composite particle field of the entire domain. What is noteworthy about this result is that the smoothness requirements are smallest for the important case $p = q = 2$ where we only require $m + 1 > d/2$. This in particular means that for $d \in \{2, 3\}$ we can chose $m = 1$.

2.4.5 Particle Dynamics

In section 2.3 we repeatedly used the word 'particle', although in a more intuitive and less rigorous meaning. Then, in the previous subsection, we introduced particle fields as quadrature rules for integrating smooth functions φ against an underlying function we are aiming to approximate. The connection between the two concepts becomes clear when looking at particle solutions to the inviscid vorticity equations. The theory of such *weak solutions* is too vast to be treated here in detail. Here, we will only give the key results. For details we refer the reader to chapter 8 of Majda and Bertozzi's book [17] or Appendix A of the book of Cottet and Koumoutsakos [48].

We will again assume that we are given a smooth, divergence free velocity field $\mathbf{u}$ that satisfies the no-through condition on ∂D. Let $\omega_h(0) = \sum_{i=1}^{N} \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i)$ be a particle field approximation of the initial vorticity field in the two-dimensional case. In the three-dimensional case we correspondingly have $\boldsymbol{\omega}_h(0) = \sum_{i=1}^{N} \boldsymbol{\Gamma}_i \delta(\mathbf{x} - \mathbf{x}_i)$. We are then looking for the solution of the inviscid vorticity equation:

$$\frac{\partial \omega_h}{\partial t} + (\mathbf{u} \cdot \nabla)\omega_h = 0 \text{ in } D \subset \mathbb{R}^2, \tag{2.54}$$

in the two-dimensional case and:

$$\frac{\partial \boldsymbol{\omega}_h}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega}_h - \boldsymbol{\omega}_h \cdot \boldsymbol{\nabla}\mathbf{u} = \mathbf{0} \text{ in } D \subset \mathbb{R}^3, \tag{2.55}$$

in the three-dimensional case. The analytic solutions of these equations are then given by modifying the particle locations and strengths according to the following set of ordinary differential equations (ODEs) [48, Theorem A.2.5]:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}(t) = \mathbf{u}(\mathbf{x}_i(t), t), \\ \dfrac{\mathrm{d}\Gamma_i}{\mathrm{d}t}(t) = 0, \end{cases} \qquad i = 1, \ldots, N, \tag{2.56}$$

in the two-dimensional case and:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}(t) = \mathbf{u}(\mathbf{x}_i(t), t), \\ \dfrac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t}(t) = \boldsymbol{\nabla}\mathbf{u}(\mathbf{x}_i(t), t) \cdot \boldsymbol{\Gamma}_i(t), \end{cases} \qquad i = 1, \ldots, N, \tag{2.57}$$

in the three-dimensional case. In other words the nodes $\mathbf{x}_i$ are convected with the flow, just like the imaginary particles from section 2.3. It is this natural treatment of convection that renders vortex methods essentially free of any numerical diffusion. Also note that the solution remains to be a particle field for $t > 0$. In practice these systems of ODEs can be discretised using explicit Runge–Kutta or multistep methods.

These results can be extended to inhomogeneous equations with right-hand sides. In particular, let us assume that we are given a particle field $f_h(t) = \sum_{i=1}^{N} F_i(t)\delta(\mathbf{x} - \mathbf{x}_i(t))$ or an equivalent vector valued version $\mathbf{f}_h$, where the particle locations $\mathbf{x}_i$ are the same as for the vorticity field ω_h. Then the analytic solution to the inhomogeneous, inviscid vorticity equation:

$$\frac{\partial \omega_h}{\partial t} + (\mathbf{u} \cdot \nabla)\omega_h = f_h \text{ in } D \subset \mathbb{R}^2, \tag{2.58}$$

is given by the modifying $\mathbf{x}_i$ and Γ_i according to the following set of ODEs:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}(t) = \mathbf{u}(\mathbf{x}_i(t), t), \\ \dfrac{\mathrm{d}\Gamma_i}{\mathrm{d}t}(t) = F_i(t), \end{cases} \qquad i = 1, \ldots, N. \tag{2.59}$$

Analogously, the solution to the three-dimensional inviscid vorticity equation:

$$\frac{\partial \boldsymbol{\omega}_h}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega}_h - \boldsymbol{\omega}_h \cdot \boldsymbol{\nabla}\mathbf{u} = \mathbf{f}_h \text{ in } D \subset \mathbb{R}^3, \tag{2.60}$$

is given by modifying $\mathbf{x}_i$ and $\boldsymbol{\Gamma}_i$ according to:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}(t) = \mathbf{u}(\mathbf{x}_i(t), t), \\ \dfrac{\mathrm{d}\boldsymbol{\Gamma}_i}{\mathrm{d}t}(t) = \boldsymbol{\nabla}\mathbf{u}(\mathbf{x}_i(t), t) \cdot \boldsymbol{\Gamma}_i(t) + \mathbf{F}_i(t), \end{cases} \qquad i = 1, \ldots, N. \tag{2.61}$$

2.5 Putting Things Together

We have now introduced the basic framework which is necessary to understand vortex methods. In section 2.1 we introduced the Navier–Stokes equations in the vorticity–velocity formulation and discussed which boundary conditions need to be prescribed. The solutions to the kinematic and dynamic parts of the equations were discussed in sections 2.2 and 2.3, respectively. In section 2.4 we described how functions can be approximated using particle fields and how the dynamic part of the flow equations can be analytically solved for particle field approximations. The basic algorithm for *inviscid* vortex particle methods can now be summarised as follows:

1. Given a computational domain $\Omega \subset D \subset \mathbb{R}^d$ and an initial vorticity field, obtain a particle field approximation $\boldsymbol{\omega}_h(0) \approx \boldsymbol{\omega}(0)$ $(d = 3)$ or $\omega_h(0) \approx \omega(0)$ $(d = 2)$, respectively.

2. Obtain a smooth approximation of the particle field $\boldsymbol{\omega}_\sigma \approx \boldsymbol{\omega}_h$ or $\omega_\sigma \approx \omega_h$, respectively.

3. Obtain a corresponding smooth velocity field $\mathbf{u}_\sigma$ using the methods described in section 2.2. That is, evaluate the Biot–Savart integral for the smooth vorticity approximation and, if $\partial D \neq \emptyset$, compute the potential flow $\mathbf{u}_\varphi$ to account for the no-through boundary condition.

4. Use the velocity approximation $\mathbf{u}_\sigma$ and apply, e. g., a Runge–Kutta scheme to advance $\boldsymbol{\omega}_h$ one step in time to equation (2.57). For the case $d = 2$ analogously advance ω_h in time according to equation (2.56).

5. Return to step 2.

The different schemes which are currently in use differ in the implementation of the individual steps. Step 1 can for example be realised using quadrature rules on quasi-uniform, shape-regular triangulations $\mathcal{G}_h$ of the computational domain as described in section 2.4.

Step 2 is called particle regularisation. This step is crucial and lies at the core of all modern vortex methods. It is required to apply the Calderón–Zygmund estimate (2.23)

in order to ensure stability and convergence of the method. In the whole-space case one can mollify the vortex particle field with certain blob functions ζ_σ of blob width σ and set $\omega_\sigma := \zeta_\sigma \star \omega_h$.

Step 3 can be carried out analytically in special cases only, as for example in vortex blob methods in the whole-space or periodic setting. In this case one can make use of the fact that the convolution operator is associative: one has $\mathbf{u}_\sigma = \mathbf{K} \star \zeta_\sigma \star \omega_h$. The corresponding smoothed kernel $\mathbf{K}_\sigma := \mathbf{K} \star \zeta_\sigma$ can often be computed analytically and then be applied to the particle field ω_h directly. In this case there are rigorous convergence proofs for vortex methods available, for example in section 2.6 of Cottet and Koumoutsakos' book [48]. Their analysis confirms that such inviscid vortex methods in $d = 2$ are free of numerical dissipation and conserve mass, linear momentum, angular momentum, and kinetic energy exactly. In the three-dimensional case the analysis of the conservation properties is significantly harder due to the additional stretching term, but numerical experiments confirm the excellent conservation properties also in this case.

2.6 Open Problems

The generic vortex method of section 2.5 consists of several steps. The current approaches for some of these steps, however, fail to yield accurate results in the presence of boundaries. In the vorticity–velocity formulation of the Navier–Stokes equations, one can clearly distinguish between a kinematic and a dynamic part. As a consequence some of the steps above can be considered in isolation. This comes with the benefit of a simplified analysis of the individual steps.

The regularisation problem (Step 2) has so far only been adequately solved in special cases like the whole-space, where one can use blob functions. Because these blobs do not adapt to the geometry's shape, they fail to give accurate regularisations near boundaries. The regularisation problem itself can be considered independent of *both* the kinematic and dynamic parts of the Navier–Stokes equations and may be seen as a link between them.

The solution of the kinematic equations (Step 3) for a given admissible vorticity field is only known analytically in the whole-space case. This problem depends on the type of regularisation used in Step 2, but is independent of the dynamical parts of the equations. Solutions for this problem can thus be used for both the viscous and inviscid equations.

In the presence of boundaries one faces with two additional problems: the Biot–Savart law is taken over a domain Ω of problem dependent shape, and can in general not be integrated analytically. Current blob methods effectively integrate over the whole-space and make use of the symmetry of the blob functions. Furthermore, the no-through boundary condition needs to be enforced by means of a boundary element method.

Alternatively, one may follow the approach of Vortex-in-Cell methods, and try to solve the Poisson equation $-\boldsymbol{\Delta}\boldsymbol{\Psi} = \boldsymbol{\omega}$ for the unknown stream function $\boldsymbol{\Psi}$, such that $\mathbf{u} = \nabla \times \boldsymbol{\Psi}$. In three-dimensional space one then faces the problem that the

boundary conditions on $\mathbf{\Psi}$ cross-couple its individual components, thereby making an implementation on general domains difficult.

As mentioned before in section 2.3, it is not immediately clear how to handle the viscous case in a particle method. For the basic heat equation a particle approximation immediately gets 'smeared out' to a smooth C^∞ function and thus ceases being a particle field. In the presence of boundaries one faces the additional problem that boundary conditions for the vorticity are in general not available and need to be obtained in an additional, separate step.

In this thesis we try to address some of these problems. Because many of them may be considered in isolation, the subsequent three chapters of this thesis may also be read separately. In chapter 3 we describe a new approach how to handle the viscous term. In order to avoid the additional complexities involved in the treatment of boundaries, we only consider the case $D = \mathbb{R}^2$. In chapter 4 we describe a general approach to tackle the regularisation problem in arbitrary bounded domains. To the best of our knowledge, this is the first truly general approach to the problem. In chapter 5 we describe simple, efficient, and accurate formulae for the evaluation of the Biot–Savart law on tetrahedra. These can be seen as a contribution to the implementation of Step 3, and improve previously known formulae. We conclude this thesis with conclusions and an outlook.

Chapter 3

Vorticity Redistribution

In this chapter we describe a general technique to approximate the application of linear differential operators to particle fields. For simplicity, we will restrict ourselves to the two-dimensional plane $D = \mathbb{R}^2$. As an important concrete example, we will apply the technique to the Laplacian, which is needed to model viscous effects. We begin with a review of some of commonly used viscous schemes and discuss their benefits and drawbacks. We then give a theoretical description of the vorticity redistribution method and prove its consistency and stability. Following the theoretical treatment, we discuss implementation issues, which are crucial for obtaining an efficient method. We conclude this chapter with numerical experiments and an outlook.

3.1 Introduction

Vortex methods were first introduced as tools to simulate inviscid, unbounded flows in two-dimensional space. As illustrated in section 2.4 this can be done by means of a particle approximation ω_h of the vorticity field ω:

$$\omega \approx \omega_h = \sum_{i=1}^{N} \Gamma_i(t)\delta(\mathbf{x} - \mathbf{x}_i(t)), \tag{3.1}$$

where $\Gamma_i \in \mathbb{R}$ denotes the weight of particle i and $\mathbf{x}_i$ refers to that particle's position. For the remainder of this chapter Γ_i will also be referred to as the *circulation* of that particle An exact solution of the inviscid vorticity equation is then obtained if one evolves $\Gamma_i(t)$, $\mathbf{x}_i(t)$ according to the following set of ordinary differential equations, for $i = 1, \ldots, N$:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{u}(t, \mathbf{x}_i(t)), \ \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} = 0, \tag{3.2}$$

where $\mathbf{u}$ refers to the exact flow velocity. In other words, particles are moved with the flow and keep their circulation. It is this natural treatment of convection that renders vortex methods essentially free of artificial viscosity. In practice, $\mathbf{u}$ will of course need to be replaced with an approximate velocity $\mathbf{u}_\sigma$, obtained for example from the Biot–Savart law and a regularised vorticity ω_σ as in section 2.2. But even in this case one can often show that the resulting scheme conserves circulation, linear momentum, angular momentum, and energy exactly. Due to the properties of the Biot–Savart integral as described in subsection 2.2.4, the continuity equation and

thereby the conservation of mass is even fulfilled in a strong, pointwise sense. Many different approaches on how to handle the viscous case have been suggested in the literature, and we will give a brief review of some of the most commonly used schemes here.

3.1.1 Viscous Splitting Algorithms

Many of the earlier approaches for handling physical viscosity belong to the class of *viscous splitting* algorithms. The core idea is to split each time step into two sub-steps. In the first sub-step one ignores viscous effects and convects particles under the absence of viscosity. In the second sub-step one then considers a pure diffusion problem and ignores convection. The advantage of this idea lies in the fact that one can now apply different approaches to each of the sub-steps and only needs to focus on the respective phenomena involved.

Practice has shown that the viscous splitting can have a stabilising effect on the discretisation; the convergence of this approach has been proven by Beale and Majda [32]. However, unless one employs more sophisticated splits, its rate of convergence is only of first order $\mathcal{O}(\nu\Delta t)$, where Δt refers to the step-width and ν to the fluid's kinematic viscosity. Note that this result holds regardless of the time-stepping schemes used for each sub-step, underlining that splitting the equation is unnatural: diffusion and convection *do happen simultaneously* and thus should not be treated one after another.

Random Walk Methods

One of the earliest approaches for a pure diffusion scheme are so-called 'random walk' methods, as introduced by Chorin [20]. These methods are based on the physical intuition that viscosity is caused by molecule collisions, which in turn result in a Brownian motion of the fluid's particles. Consequently, the idea is to randomly move particles according to a specific probability distribution. Due to the random nature of this process, these methods are called indeterministic. A detailed description and a proof of its convergence can for example be found in Beale and Majda's book [17, Chapter 6]. While being intuitive, these methods generally suffer from very slow convergence rates.

Core Spreading Methods

Core spreading methods rely on a particular approach for regularising a given particle field ω_h. In so-called blob methods this regularisation is achieved by mollifying the particle field ω_h with a carefully chosen blob-function ζ_σ, where σ refers to the size of the blob's core. The smoothed vorticity field is then given by $\omega_\sigma = \omega_h \star \zeta_\sigma$. This effectively corresponds to replacing δ in equation (3.1) with ζ_σ. If one chooses so-called Gaussian blobs, ζ_σ corresponds to a scaled heat kernel, i. e., a solution to the heat equation. Advancing this solution in time then corresponds to increasing σ. The original approach has been shown to be inconsistent [62], but this could later be fixed by Rossi [63].

The gradual increase of σ causes the solution to get more and more blurred over time and thus further increases the problems blob methods are already facing in the presence of boundaries. Some kind of regridding scheme is required to prevent σ from becoming too large to capture the flow's details.

Vorticity Redistribution Method

The vorticity redistribution method (VRM) by Shankar and van Dommelen [64] can be interpreted as a computed finite-difference stencil which solves the heat-equation for a given time-step Δt. This approach makes no assumptions on the particle geometry or the regularisation scheme used. If necessary, new particles are inserted in to the flow in areas without enough particles. For this a suitable heuristic needs to be used. The fact that stencils are computed on-the-fly makes the method completely mesh-free. Our approach described in this chapter will build on this idea.

3.1.2 Integral Based Algorithms

The approaches described in this subsection are based on the assumption that the particle field was initialised with the help of a quadrature rule as described in section 2.4. Keeping the quadrature weights w_i fixed, one may expect that a slight change in the positions of the quadrature nodes $\mathbf{x}_i$ will still yield a good quadrature rule for the computational domain. The idea of the following approaches is based on this assumption. As the particles get convected with the flow, this assumption will cease to hold after some limited amount of time. Subsequently the approaches presented here rely on a frequent reinitialisation of the particle field, which introduces considerable amounts of artificial viscosity.

Analytic Solution Approach

The first of these approaches [48, Section 5.3] also belongs to the class of viscous splitting algorithms and is based on the fact that in the whole-space case an analytic solution of the heat equation is available, namely:

$$\omega(t + \Delta t, \mathbf{x}) = \int_{\mathbb{R}^2} H(\nu \Delta t, \mathbf{x} - \mathbf{y}) \omega(t, \mathbf{y}) \, \mathrm{d}\mathbf{y}, \tag{3.3}$$

where H is the heat kernel:

$$H(\tau, \mathbf{x} - \mathbf{y}) = \frac{1}{4\pi\tau} \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{4\tau}\right), \tag{3.4}$$

which is smooth for $\tau > 0$. By replacing ω with ω_h in the above formula, one then obtains a smooth approximation of the diffused vorticity at the next time-step. This smooth approximation can then be turned into a particle field by reusing the weights w_i used in the original quadrature rule with the updated current particle positions $\mathbf{x}_i$. In addition to the need to frequently reinitialize the particle field, due to the viscous splitting this scheme is also limited to first order accuracy in time.

Particle Strength Exchange

In the method of particle strength exchange (PSE), the Laplace operator itself is approximated by an integral, instead of a solution of the heat equation [48, Section 5.4]. This results in a smooth approximation of the Laplacian of the particle field. This smooth approximation can then in turn be turned into a particle field by reusing the original quadrature weights w_i with the updated particle locations $\mathbf{x}_i$. This frees the method of the viscous splitting and potentially allows for arbitrary order discretisations in time. As mentioned before, frequent reinitialisation is required to ensure the method's accuracy.

An approach which potentially bypasses this necessity has recently been published under the name discretisation corrected particle strength exchange (DC-PSE) [65]. This approach requires the solution of small scale systems for each particle to account for the distortion of the particle field, and is in some ways similar to the approach we will follow here. It lacks, however, a proof of its stability.

3.1.3 Discussion and Outline

With the exception of the DC-PSE, all off the above approaches either require a frequent particle reinitialisation or make use of the viscous splitting. The DC-PSE on the other hand, is similar in nature to the vorticity redistribution method, but lacking a stability proof.

In this chapter we describe a method for approximating the application of linear differential operators to particle fields, in a manner similar to the vorticity redistribution method (VRM). As a concrete example we will analyse its application to the Laplace operator to simulate diffusion. This allows us to avoid the viscous splitting and to treat both diffusion and convection simultaneously. The spatial consistency of our method is proven. We then consider the case of pure diffusion in combination with the forward Euler method and derive sharp a-priori and a-posteriori bounds on the step-width. This analysis in the absence of convection is justified, as the convective part of the equations is known to be stable independent of the step-width [31]. The resulting a-priori bound is—apart from a constant—identical to the classical stability condition for the five-point central-difference stencil, underlining the interpretation of our method as a computed finite-difference method. Finally, we show that the method conserves circulation, linear, and angular momentum.

In the original description of the VRM it was suggested to ignore particles in the diffusive process if their circulation was below a certain threshold. Choosing a low threshold does yield accurate discretisations, however, the choice of its value seemed rather arbitrary. We propose a new strategy preventing excessive growth in the number of particles while maintaining the order of consistency. Based on results by Seibold [66, 67], we further introduce the new concept of *small neighbourhoods* which significantly reduces the computational cost of the method. The resulting scheme keeps all of the benefits of the original VRM while not relying on viscous splitting or arbitrary thresholds. We conclude with numerical examples illustrating efficiency and convergence

of the method in the purely diffusive, as well as in the convective case.

3.2 Derivatives of Particle Fields

Let L denote a linear differential operator with constant coefficients, e. g., $L = \Delta$ for the Laplacian, or $L = \mathrm{Id}$ for the identity. The application of such operators to particle fields ω_h can then be defined using the concept of distributional derivatives [68, Section 1.60]. The distribution $L\omega_h$ is defined by:

$$\langle L\omega_h, \varphi \rangle := \langle \omega_h, L^* \varphi \rangle \qquad \forall \varphi \in C_0^\infty(\mathbb{R}^2). \tag{3.5}$$

Here, L^* denotes the so-called formal adjoint of L. The Laplacian and the identity operators are symmetric and in this case we have $L = L^*$.

In section 2.4 the error of particle approximations was quantified with the help of negative index Sobolev norms $\|\cdot\|_{W^{-m,q}(\mathbb{R}^2)}$. It thus makes sense to replace $\varphi \in C_0^\infty(\mathbb{R}^2)$ with the larger test-space $W_0^{m+o,p}(\mathbb{R}^2) = W^{m+o,p}(\mathbb{R}^2)$, where $p \in [1, \infty]$ is the index conjugate to q, such that $1/p + 1/q = 1$, and o denotes the order of the differential operator L, i. e., $o = 2$ for the Laplacian, and $o = 0$ for the identity.

Our aim is now to find a particle field approximation $L_h\omega_h$:

$$L_h\omega_h := \sum_{i=1}^{N} \lambda_i \delta(\mathbf{x} - \mathbf{x}_i) \tag{3.6}$$

of $L\omega_h$ such that:

$$\langle L_h\omega_h, \varphi \rangle \approx \langle \omega_h, L^* \varphi \rangle \qquad \forall \varphi \in W^{m+o,p}(\mathbb{R}^2). \tag{3.7}$$

The question that then automatically arises is how to choose the weights λ_i such that the above assertion holds. For the case $p = \infty$ and $q = 1$ this question will be answered in the following two sections. The key observation is that under certain additional assumptions on the weights λ_i, a converging scheme is obtained if one enforces $\langle L\omega_h, \varphi \rangle = \langle L_h\omega_h, \varphi \rangle$ for all polynomial test-functions φ of degree o. Enforcing the relation for polynomials of higher degrees will then increase the convergence order.

As a side note we point out that the particle field $L_h\omega_h$ may—but does not have to—use the same particle locations as the original field ω_h. The latter option could for example be used in areas where too many particles have clustered together, like stagnation points. Setting $L = I$, and ω_h to a collection of 'superfluous' particles, one may try to approximate this set with help of the remaining particles.

Reducing the number of particles is primarily an efficiency concern and not a question of stability or accuracy. We will thus focus on the Laplace operator in the rest of this chapter. Choosing the same particle locations for $L_h\omega_h$ and ω_h then naturally fits into the framework of subsection 2.4.5: viscous flows may be simulated by setting the right-hand side to $f_h := \nu\Delta_h\omega_h$.

3.3 Description of the Method

3.3.1 Preliminaries

We assume that we are given a particle approximation $\omega_h = \sum_{i=1}^N \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i)$ of a vorticity field $\omega \in W^{m,1}(\mathbb{R}^2)$, satisfying an error bound of the form $\|\omega_h - \omega\|_{W^{-m,1}} \leq Ch^m \|\omega\|_{W^{m,1}}$ for some integer $m \geq 1$. The test functions $\varphi \in W^{m,\infty}(\mathbb{R}^2)$ and all of their derivatives up to order $m-1$ are bounded and continuous. These functions actually fulfil the even stronger notion of Lipschitz continuity [60, Exercises 1.x.14 and 1.x.15] and thus particle approximations are well defined.

For later reference, note that the norm of such a particle approximation may be upper bounded by $\|(\Gamma_i)\|_{l^1}$:

$$\begin{aligned} \|\omega_h\|_{W^{-m,1}} &= \sup_{\varphi \in W^{m,\infty}} \frac{|\langle \omega_h, \varphi \rangle|}{\|\varphi\|_{W^{m,\infty}}} \leq C_{\text{emb}} \sup_{\varphi \in W^{m,\infty}} \frac{\sum_{i=1}^N |\Gamma_i| \, |\varphi(\mathbf{x}_i)|}{\|\varphi\|_C} \\ &\leq C_{\text{emb}} \sup_{\varphi \in W^{m,\infty}} \frac{\|\varphi\|_C \sum_{i=1}^N |\Gamma_i|}{\|\varphi\|_C} = C_{\text{emb}} \|(\Gamma_i)\|_{l^1}, \end{aligned} \tag{3.8}$$

where $\langle \cdot, \cdot \rangle$ refers to the dual pairing, $\| \cdot \|_C$ is the supremum norm and C_{emb} denotes the Sobolev embedding constant. This inequality will allow us to infer stability in the $W^{-m,1}$-norm by bounding the l^1-norm of the circulations later on.

If we additionally assume that the particle field stems from a quadrature rule of exactness degree $m-1$ with positive weights as described in section 2.4, the quantity $\|(\Gamma_i)\|_{l^1}$ allows for a natural interpretation: it is the quadrature approximation of the L^1-norm of the vorticity, i. e., $\|\omega\|_{L^1} \approx \|(\Gamma_i)\|_{l^1}$.

3.3.2 Moment Conditions

Our aim is to approximate the Laplacian of ω_h with a particle field using the same particle locations. We thus chose the following ansatz:

$$\Delta_h \omega_h := \sum_{i=1}^N \sum_{j=1}^N f_{ij} \Gamma_i \delta(\mathbf{x} - \mathbf{x}_j). \tag{3.9}$$

Here $f_{ij}\Gamma_i$ may be interpreted as the rate at which circulation is diffused from particle i to particle j. The question that then needs to be answered is how the values of f_{ij} need to be chosen such that this is an accurate approximation. The conditions that the f_{ij} need to fulfil are called the *moment conditions* and are described in this section.

In order to specify these conditions, we will define be particle neighbourhoods. The neighbourhood $\mathcal{N}_i$ of particle i is defined as follows:

$$\mathcal{N}_i := \{j \in 1, \ldots, N : rh \leq |\mathbf{x}_i - \mathbf{x}_j| \leq Rh\} \cup \{i\}. \tag{3.10}$$

where $R > r > 0$ are fixed, user-defined parameters. The original VRM formulation does not include the lower bound r. In our analysis we show that both bounds are

required to control the error: the upper bound limits the cut-off error of the expansions used, while the lower bound is needed for stability. For $j \in \mathcal{N}_i$ the values f_{ij} are chosen such that certain moment conditions are fulfilled. For $j \notin \mathcal{N}_i$ we define $f_{ij} = 0$.

As will be seen later on, depending on the geometry of the particle locations, these moment conditions do not always have a solution. This is for example the case for an essentially empty neighbourhood $\mathcal{N}_i = \{i\}$. Note, however, that we can always add new particles of zero strength to the field: adding a new particle i with $\Gamma_i = 0$ to ω_h leaves the particle field unchanged. For such empty particles of zero strength one obviously has $\Gamma_i = 0 \Longrightarrow f_{ij}\Gamma_i = 0$, i. e., the value of f_{ij} is arbitrary and can safely be defined as zero, too. We will make use of this fact and insert new particles according to a certain heuristic, which will be described later in this chapter. This way we can assume that the moment conditions always allow for a solution. Circulation will then be diffused from particles to their potentially empty neighbours and thereby be spread out in space, which also is in accordance with the physical intuition of diffusive processes.

At the core of our method lies the computation of the values f_{ij} for every i and $j \in \mathcal{N}_i$. The test functions φ can be approximated by suitable polynomials. For a consistent approximation we will then require that for all constant, linear, and quadratic polynomial test-functions φ the error vanishes. For greater accuracy one may require corresponding higher order polynomials to vanish. A detailed derivation of the resulting equations is given in section 3.4.1.

As will be seen later on, non-negativity of stencils is a sufficient criterion to ensure consistency and stability. In addition to that, such stencils possess many more desirable properties, as described by Seibold [66, 67, 69]. A stencil is called *non-negative* if it fulfils $f_{ij} \geq 0$ for all $j \neq i$. Somewhat less precise, such stencils are also called *positive*. Unfortunately, as will also be shown in the analysis section 3.4.5, non-negative stencils cannot fulfil the moment equations of fourth order. Unless one gives up on non-negativity and the resulting guarantees, the method's accuracy is hence limited to second order.

The moment conditions are most easily expressed using multi-index notation. Defining the vector $\mathbf{r}_{ij}$:

$$\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i, \tag{3.11}$$

and denoting its Cartesian components by $\mathbf{r}_{ij}^x$ and $\mathbf{r}_{ij}^y$, respectively, for $\mathcal{O}(h^m)$ accuracy, we pose the following conditions:

$$\sum_{j=1}^{N} f_{ij}\mathbf{r}_{ij}^x\mathbf{r}_{ij}^x = 2, \quad \sum_{j=1}^{N} f_{ij}\mathbf{r}_{ij}^y\mathbf{r}_{ij}^y = 2, \quad \sum_{j=1}^{N} f_{ij}\mathbf{r}_{ij}^x\mathbf{r}_{ij}^y = 0, \tag{3.12}$$

and for all other error terms with multi-index α:

$$\sum_{j=1}^{N} f_{ij}\mathbf{r}_{ij}^\alpha = 0, \qquad 0 \leq |\alpha| \leq m+1,\ |\alpha| \neq 2. \tag{3.13}$$

Because we have $\mathbf{r}_{ii} \equiv \mathbf{0}$, only the equation for $\alpha = (0,0)$ depends on f_{ii}, yielding:

$$f_{ii} = -\sum_{j \neq i} f_{ij}. \tag{3.14}$$

If one chooses for example $m = 1$ or $m = 2$, we consequently have to solve a system consisting of five or nine moment conditions, respectively. Together with the positivity constraint, for every particle i, this linear system can be rewritten in matrix–vector notation:

$$\mathsf{V}_i \mathsf{f}_i = \mathsf{b},\ \mathsf{f}_i \geq 0. \tag{3.15}$$

Here, f_i is the vector of coefficients f_{ij}, $i \neq j$, b is the vector that contains only zero entries except for the two '2'-entries at $\alpha = (2, 0)$ and $\alpha = (0, 2)$, and V_i is the Vandermonde matrix, with rows for each multi-index $1 \leq |\alpha| \leq m + 1$ and columns j for each particle $j \in \mathcal{N}_i \setminus \{i\}$:

$$V_{\alpha,j} = \mathbf{r}_{ij}^{\alpha}. \tag{3.16}$$

In order to obtain scaling independent of h, for a numerical implementation it is beneficial and straightforward to rewrite these conditions for the normalised vectors $\mathbf{r}_{ij}/h$. In section 3.5 we describe how to solve these equations and how to ensure that non-negative stencils exist.

3.3.3 A One-dimensional Example

In order to illustrate the meaning of the moment conditions, we will consider the following one-dimensional example. Using a linear combination of particles at positions $x = -h, 0$, and $\tilde{h}$, we wish to approximate $\frac{\mathrm{d}^2}{\mathrm{d}x^2}\delta(x)$, i. e., we want to find coefficients a, b, and c such that:

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\delta(x) \approx a\delta(x + h) + b\delta(x - \tilde{h}) + c\delta(x). \tag{3.17}$$

In other words, for every smooth test-function φ one should have:

$$\varphi''(0) \approx a\varphi(-h) + b\varphi(\tilde{h}) + c\varphi(0). \tag{3.18}$$

Expanding φ as a Taylor series around 0 one obtains:

$$\begin{aligned}
\varphi(-h) &\approx \varphi(0) - h\frac{\partial\varphi}{\partial x}(0) + \frac{h^2}{2}\frac{\partial^2\varphi}{\partial x^2}(0),\\
\varphi(0) &= \varphi(0),\\
\varphi(+\tilde{h}) &\approx \varphi(0) + \tilde{h}\frac{\partial\varphi}{\partial x}(0) + \frac{\tilde{h}^2}{2}\frac{\partial^2\varphi}{\partial x^2}(0).
\end{aligned}$$

From this we obtain the following conditions on the coefficients:

- $a + b + c = 0$ in order for the term $\varphi(0)$ to vanish.
- $\tilde{h}b - ha = 0$ for the term first order derivative, and
- $\frac{h^2}{2}a + \frac{\tilde{h}^2}{2}b = 1$ for the second order derivative.

This system of equations is the one-dimensional equivalent of the moment equations described above. In this particular case it has a unique solution:

$$a = \frac{2}{h^2 + h\tilde{h}}, \quad b = \frac{2}{h\tilde{h} + \tilde{h}^2}, \quad c = -(a + b). \tag{3.19}$$

Let us now for example assume that one would have $\tilde{h} = (1 + \varepsilon)h$. In this case the number ε can be seen as a measure of the asymmetry of the particle distribution. The upper and lower bounds Rh and rh from the definition of the particle neighbourhoods in the previous section enforce that this number remains bounded. Including higher order terms in the Taylor expansions, one obtains:

$$\varphi''(0) - (a\varphi(-h) + b\varphi(\tilde{h}) + c\varphi(0)) = \mathcal{O}(\varepsilon h + h^2). \tag{3.20}$$

Thus—strictly speaking—the resulting method is first order accurate with respect to h. For small values of ε one may however expect the higher order term to be dominant and actually observe second order convergence for practical choices of h.

3.4 Analysis

3.4.1 Consistency

We begin by deriving certain key properties of positive stencils.

Lemma 3.1. *Let (f_{ij}) be a positive stencil that satisfies the moment conditions* (3.15)*. One then has the following sharp estimates:*

$$f_{ii} = -\sum_{j \neq i} f_{ij} \leq 0, \quad 4(Rh)^{-2} \leq \sum_{j \neq i} f_{ij} \leq 4(rh)^{-2}. \tag{3.21}$$

Proof. The first equality directly follows equation (3.14) and the fact that for positive stencils $f_{ij} \geq 0$ for $i \neq j$. Now note that one obtains $\sum_{j \neq i} |\mathbf{r}_{ij}|^2 f_{ij} = 4$ by adding the moment equations for $\alpha = (2, 0)$ and $\alpha = (0, 2)$. By the definition of the particle neighbourhoods $\mathcal{N}_i$ one furthermore has $rh \leq |\mathbf{r}_{ij}| \leq Rh$. We thus obtain with help of the positivity: $4 = \sum_{j \neq i} |\mathbf{r}_{ij}|^2 f_{ij} \geq (rh)^2 \sum_{j \neq i} f_{ij}$. The lower bound follows analogously. □

The upper bound is crucial for the consistency and stability proofs below and cannot be obtained from the moment conditions without any further assumptions on the stencils, such as positivity.

Theorem 3.2 (Consistency). *Let $\Delta_h \omega_h$ be defined as in equation* (3.9)*, where the f_{ij} form a positive stencil satisfying the moment conditions as defined above. One then has:*

$$\|\Delta\omega_h - \Delta_h\omega_h\|_{W^{-(m+2),1}} \leq C\left(\frac{R}{r}\right)^2 (Rh)^m \|(\Gamma_i)\|_{l^1},$$

where the constant C is independent of h and ω_h.

Proof. For arbitrary $\varphi \in W^{m+2,\infty}$ one has with the help of Hölder's inequality:

$$\langle \Delta\omega_h - \Delta_h\omega_h, \varphi\rangle = \sum_{i=1}^{N} \Gamma_i \Big(\Delta\varphi(\mathbf{x}_i) - \sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j)\Big) \leq \|(\Gamma_i)\|_{l^1} \|\Delta\varphi(\mathbf{x}_i) - \sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j)\|_{l^\infty}. \tag{3.22}$$

Let us thus consider the value of the $\|\cdot\|_{l^\infty}$-part on the right. Due to the definition of the particle neighbourhoods $\mathcal{N}_i$, for each i the sum $\sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j)$ only evaluates φ within the ball $B_{Rh}(\mathbf{x}_i)$ of radius Rh centred at $\mathbf{x}_i$; for the particles outside this sphere one has by definition $f_{ij} = 0$. According to the Bramble–Hilbert lemma [60, Lemma (4.3.8)], there exists a polynomial $\mathcal{P}\varphi$ of total degree less than $m+2$ such that:

$$|\varphi - \mathcal{P}\varphi|_{W^{l,\infty}(B_{Rh}(\mathbf{x}_i))} \leq C(Rh)^{m+2-l} |\varphi|_{W^{m+2,\infty}(B_{Rh}(\mathbf{x}_i))} \quad \forall l \in \mathbb{N}_0, l \leq m+2. \tag{3.23}$$

Because $\mathcal{P}\varphi$ is a polynomial, it equals its Taylor series around $\mathbf{x}_i$ and we obtain:

$$\mathcal{P}\varphi(\mathbf{x}_j) = \sum_{|\alpha|\leq m+1} \frac{\mathbf{r}_{ij}^{\alpha}}{\alpha!} D^{\alpha}\mathcal{P}\varphi(\mathbf{x}_i). \tag{3.24}$$

Using this relation, one quickly sees that the moment conditions were chosen such that:

$$\Delta\mathcal{P}\varphi(\mathbf{x}_i) = \sum_{j=1}^{N} f_{ij}\mathcal{P}\varphi(\mathbf{x}_j) \tag{3.25}$$

for all polynomials $\mathcal{P}\varphi$ of total degree less than $m+2$. Thus:

$$\Delta\varphi(\mathbf{x}_i) - \sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j) = \Delta(\varphi - \mathcal{P}\varphi)(\mathbf{x}_i) - \sum_{j=1}^{N} f_{ij}(\varphi - \mathcal{P}\varphi)(\mathbf{x}_j). \tag{3.26}$$

By the Bramble–Hilbert lemma we then obtain:

$$|\Delta(\varphi - \mathcal{P}\varphi)(\mathbf{x}_i)| \leq C|\varphi - \mathcal{P}\varphi|_{W^{2,\infty}(B_{Rh}(\mathbf{x}_i))} \leq C(Rh)^m \|\varphi\|_{W^{m+2,\infty}}, \tag{3.27}$$

$$|(\varphi - \mathcal{P}\varphi)(\mathbf{x}_j)| \leq C|\varphi - \mathcal{P}\varphi|_{W^{0,\infty}(B_{Rh}(\mathbf{x}_i))} \leq C(Rh)^{m+2} \|\varphi\|_{W^{m+2,\infty}}. \tag{3.28}$$

Using lemma 3.1 and Hölder's inequality we then obtain:

$$\begin{aligned}\Big|\sum_{j=1}^{N} f_{ij}(\varphi - \mathcal{P}\varphi)(\mathbf{x}_j)\Big| &\leq C(Rh)^{m+2}\|\varphi\|_{W^{m+2,\infty}(\mathbb{R}^2)}\|(f_{ij})_j\|_{l^1} \\ &\leq C\Big(\frac{R}{r}\Big)^2 (Rh)^m \|\varphi\|_{W^{m+2,\infty}(\mathbb{R}^2)}.\end{aligned} \tag{3.29}$$

Thus for each i one has:

$$\Big|\Delta\varphi(\mathbf{x}_i) - \sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j)\Big| \leq C\Big(\frac{R}{r}\Big)^2 (Rh)^m \|\varphi\|_{W^{m+2,\infty}}, \tag{3.30}$$

with a constant C independent of i and h and the claim follows. □

3.4.2 Stability for the Heat Equation

In our next step we investigate the stability of Euler's method in combination with our spatial discretisation. As we introduced an approximation for the viscous term, it makes sense to ignore convective effects and assume $\mathbf{u} = \mathbf{0}$. We thus consider the heat equation with a positive diffusion coefficient $\nu > 0$:

$$\frac{\partial \omega}{\partial t} = \nu \Delta \omega. \tag{3.31}$$

For a given particle approximation ω_h this leads to the semi-discrete system:

$$\frac{\partial \omega_h}{\partial t} = \nu \Delta_h \omega_h, \tag{3.32}$$

with Δ_h as defined before. Following subsection 2.4.5 with $f_h = \Delta_h \omega_h$ and $\mathbf{u} \equiv \mathbf{0}$ the solution of this system then can then be given by:

$$\frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} = \sum_{j=1}^{N} f_{ji} \Gamma_j, \qquad i = 1, \dots, N. \tag{3.33}$$

We will discretise this system of ordinary differential equations using Euler's method. This method is only first order accurate in time, but higher order schemes can be constructed by combining several Euler steps, as for example in so-called SSP-stable schemes [70]. The classical Runge–Kutta method (RK4) is not such a scheme, but our numerical experiments at the end of this chapter showed no instabilities.

Introducing the vector $\mathsf{\Gamma} \in \mathbb{R}^N$ with components Γ_i, and the matrix $\mathsf{F} \in \mathbb{R}^{N \times N}$ consisting of components f_{ij}, the resulting scheme can be written as:

$$\mathsf{\Gamma}^{n+1} = \underbrace{(\mathsf{I} + \nu \Delta t \mathsf{F}^\top)}_{=:\mathsf{C}} \mathsf{\Gamma}^n, \tag{3.34}$$

Here, $\Delta t > 0$ is the step width, I is the identity matrix, and n and $n+1$ denote time-steps n and $n+1$, respectively. The fact that C involves the transpose $\mathsf{F}^\top$ of F is a reminder of the transposed operator L^* from the definition of particle field derivatives in section 3.2. Although the Laplacian is a symmetric operator, due to the typically asymmetric particle distributions the matrix F will usually not be symmetric.

As shown in theorem 3.2, the consistency error can be bounded by $Ch^m \|\mathsf{\Gamma}\|_1$. In order to bound $\|\mathsf{\Gamma}\|_1$ for all times it is therefore sufficient to require that $\|\mathsf{C}\|_1 \leq 1$. Note that due to equation (3.8), this implies that $\|\omega_h\|_{W^{-m,1}}$ remains bounded as well. The following theorem will show that positive stencils are not only sufficient but also necessary to obtain a scheme that fulfils $\|\mathsf{C}\|_1 \leq 1$.

Theorem 3.3 (Stability). *Let f_{ij} denote a stencil that satisfies the moment conditions* (3.15)*. Then the matrix C as defined above satisfies $\|\mathsf{C}\|_1 = 1$ if and only if the stencil is positive: $f_{ii} \leq 0$, $f_{ij} \geq 0$ $(i \neq j)$, and the time-step satisfies the bound $\nu \Delta t \leq -f_{ii}^{-1}$ for all $i = 1, \dots, N$. For larger Δt or non-positive stencils one always has $\|\mathsf{C}\|_1 > 1$.*

Proof. One has:

$$\|\mathsf{C}\|_1 = \max_j \sum_{i=1}^{N} |C_{ij}| = \max_j |1 + \nu\Delta t f_{jj}| + \nu\Delta t \sum_{i\neq j} |f_{ji}|. \tag{3.35}$$

Thus $\|\mathsf{C}\|_1 \leq 1 \implies f_{jj} \leq 0$. Let us consider the case that $(1 + \nu\Delta t f_{jj}) \geq 0$, i. e., $\nu\Delta t \leq -f_{jj}^{-1}$. After substituting $f_{jj} = -\sum_{i\neq j} f_{ji}$ we obtain for each j:

$$\sum_{i=1}^{N} |C_{ij}| = 1 - \nu\Delta t \sum_{i\neq j} f_{ji} + \nu\Delta t \sum_{i\neq j} |f_{ji}|. \tag{3.36}$$

Thus, we have $\|\mathsf{C}\|_1 \leq 1$ if and only if for all j:

$$\sum_{i\neq j} |f_{ji}| \leq \sum_{i\neq j} f_{ji} \iff f_{ji} \geq 0. \tag{3.37}$$

For positive stencils both sides are equal, and thus $\|\mathsf{C}\|_1 = 1$.

In the opposite case we have $(1 + \nu\Delta t f_{jj}) < 0$, i. e., $\nu\Delta t > -f_{jj}^{-1}$. After using the same substitution for f_{jj} we then obtain for each j:

$$\sum_{i=1}^{N} |C_{ij}| = -1 + \nu\Delta t \sum_{i\neq j} (f_{ij} + |f_{ij}|). \tag{3.38}$$

Assume we would have $\|\mathsf{C}\|_1 \leq 1$. We then would have for all j:

$$\sum_{i\neq j} (f_{ij} + |f_{ij}|) \leq \frac{2}{\nu\Delta t}. \tag{3.39}$$

But note that we have:

$$\sum_{i\neq j} (f_{ij} + |f_{ij}|) \geq 2 \sum_{i\neq j} f_{ij} = 2 f_{jj}, \tag{3.40}$$

and thus:

$$2 f_{jj} \leq \frac{2}{\nu\Delta t} \iff \nu\Delta t \leq -f_{jj}^{-1}, \tag{3.41}$$

which is a direct contradiction to our assumption on the time-step. □

Theorem 3.3 gives us an easy a-posteriori bound which can readily be implemented. This allows us to optimally choose the step-width in a computer program. In higher-order Runge–Kutta schemes it is hard to predict the values f_{ii} for intermediate stages. Thus, again employing Lemma 3.1, the following a-priori bound is useful:

$$\Delta t \leq \frac{(rh)^2}{4\nu}. \tag{3.42}$$

Note that this closely resembles the classical stability condition for the five-point finite-difference stencil, highlighting the similarity between the two methods. The fact that we can only achieve $\|\mathsf{C}\|_1 = 1$, as opposed to $\|\mathsf{C}\|_1 < 1$, can be seen as a consequence of the fact that our method conserves circulation, as will be shown in the next section.

3.4.3 Conservation Properties for the Navier–Stokes Equations

The excellent conservation properties of vortex methods have always been one of their strong points. While the conservation of mass is usually fulfilled in a strong, point-wise sense due to the Biot–Savart law, vortex methods usually also conserve circulation, linear as well as angular momentum. In this section we will show that this is also applies in the viscous case when we apply the vorticity redistribution method. We will thus consider the following semi-discrete system of coupled ordinary differential equations:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} &= \mathbf{u}_\sigma(\mathbf{x}_i), \\ \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} &= \nu \sum_{j=1}^{N} f_{ji}\Gamma_j, \end{aligned} \tag{3.43}$$

where the fractions f_{ij} depend on the particle positions. The velocity field $\mathbf{u}_\sigma$ is obtained from the vorticity after an appropriate smoothing process. In unbounded or periodic flows it is customary to smooth the particle field ω_h with a blob function ζ_σ by mollification. Setting $\omega_\sigma := \zeta_\sigma \star \omega_h$ and assuming $\mathbf{u}_\infty = \mathbf{0}$ the approximate velocity can then be obtained using the Biot–Savart law: $\mathbf{u}_\sigma = \mathbf{K} \star \omega_\sigma$, where $\mathbf{K}$ is the Biot–Savart kernel from equation (2.12). Noting that the convolution operator is associative, this can be rewritten as $\mathbf{u}_\sigma = \mathbf{K} \star \zeta_\sigma \star \omega_h = \mathbf{K}_\sigma \star \omega_h$, with $\mathbf{K}_\sigma := \mathbf{K} \star \zeta_\sigma$. Thus:

$$\mathbf{u}_\sigma(\mathbf{x}_i) = \sum_{j=1}^{N} \mathbf{K}_\sigma(\mathbf{r}_{ji})\Gamma_j. \tag{3.44}$$

For the commonly used radial blob functions ζ_σ, the resulting smoothed kernel $\mathbf{K}_\sigma$ is an odd function that is orthogonal to its argument:

$$\mathbf{K}_\sigma(\mathbf{r}_{ij}) = -\mathbf{K}_\sigma(\mathbf{r}_{ji}), \quad \mathbf{K}_\sigma(\mathbf{r}_{ij}) \cdot \mathbf{r}_{ij} \equiv 0. \tag{3.45}$$

With the help of these properties, we are going to investigate the following quantities:

- Circulation: $I_0 := \int_{\mathbb{R}^2} \omega \quad \mathrm{d}\mathbf{x} = \sum_{i=1}^{N} \Gamma_i$,
- Linear Momentum: $\mathbf{I}_1 := \int_{\mathbb{R}^2} \omega\mathbf{x} \;\; \mathrm{d}\mathbf{x} = \sum_{i=1}^{N} \Gamma_i \mathbf{x}_i$,
- Angular Momentum: $I_2 := \int_{\mathbb{R}^2} \omega\mathbf{x}^2 \, \mathrm{d}\mathbf{x} = \sum_{i=1}^{N} \Gamma_i \mathbf{x}_i^2$.

The conservation laws for these quantities read [17, 71]:

$$\frac{\mathrm{d}I_0}{\mathrm{d}t} = 0, \;\; \frac{\mathrm{d}\mathbf{I}_1}{\mathrm{d}t} = \mathbf{0}, \;\; \frac{\mathrm{d}I_2}{\mathrm{d}t} = 4\nu I_0. \tag{3.46}$$

Note that these quantities are moments of vorticity and thus are closely linked to the moment conditions (3.13) and (3.12). This close link will allow us to show that the semi-discrete equations (3.43) fulfil the conservation laws (3.46) exactly.

THEOREM 3.4 (CONSERVATION OF MOMENTUM). *Let the smoothed Biot–Savart kernel $\mathbf{K}_\sigma$ be an odd function that is orthogonal to its argument, and let f_{ij} be a stencil that satisfies the moment conditions* (3.15)*. Then vorticity field described by the system of ODEs* (3.43) *conserves circulation as well as linear and angular momentum.*

Proof. The proof is similar to the inviscid case [48]. For circulation we immediately obtain:

$$\frac{\mathrm{d}I_0}{\mathrm{d}t} = \sum_{i=1}^{N} \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} = \nu \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ji}\Gamma_j = \nu \sum_{j=1}^{N} \Gamma_j \underbrace{\sum_{i=1}^{N} f_{ji}}_{=0} = 0. \tag{3.47}$$

For linear momentum we have:

$$\frac{\mathrm{d}\mathbf{I}_1}{\mathrm{d}t} = \sum_{i=1}^{N} \Gamma_i \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} + \sum_{i=1}^{N} \mathbf{x}_i \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t}. \tag{3.48}$$

For the first part we use the oddity of the kernel $\mathbf{K}_\sigma(\mathbf{r}_{ji}) = -\mathbf{K}_\sigma(\mathbf{r}_{ij})$ and obtain after an exchange the order of summation:

$$\sum_{i=1}^{N} \Gamma_i \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{K}_\sigma(\mathbf{r}_{ji})\Gamma_i\Gamma_j = -\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{K}_\sigma(\mathbf{r}_{ji})\Gamma_i\Gamma_j. \tag{3.49}$$

Thus, this part of the sum equals its negative and therefore is zero. For the second part we have using the moment conditions:

$$\nu \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ji}\Gamma_j\mathbf{x}_i = \nu \sum_{j=1}^{N} \Gamma_j \Bigg(\underbrace{\sum_{i=1}^{N} f_{ji}\mathbf{r}_{ji}}_{=\mathbf{0}} + \mathbf{x}_j \underbrace{\sum_{i=1}^{N} f_{ji}}_{=0} \Bigg) = \mathbf{0}. \tag{3.50}$$

Lastly, for the angular momentum we obtain:

$$\frac{\mathrm{d}I_2}{\mathrm{d}t} = \sum_{i=1}^{N} 2\Gamma_i\mathbf{x}_i \cdot \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} + \sum_{i=1}^{N} \mathbf{x}_i^2 \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t}. \tag{3.51}$$

For the first sum we have:

$$2\sum_{i=1}^{N}\sum_{j=1}^{N} \Gamma_i\Gamma_j \mathbf{K}_\sigma(\mathbf{r}_{ji}) \cdot \mathbf{x}_i. \tag{3.52}$$

By writing $\mathbf{x}_i = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j) + \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)$ this sum again splits up into two parts. Using the oddness property of $\mathbf{K}_\sigma$ and exchanging the indices as above, the first part is zero. For the second part we use that $\mathbf{K}_\sigma$ is orthogonal to its argument $\mathbf{K}_\sigma(\mathbf{r}_{ji}) \cdot \mathbf{r}_{ji} \equiv 0$.

Finally, we have $\mathbf{x}_i^2 = \mathbf{r}_{ji}^2 + 2\mathbf{x}_i^x\mathbf{x}_j^x + 2\mathbf{x}_i^y\mathbf{x}_j^y - \mathbf{x}_j^2$ and thus:

$$\sum_{i=1}^{N} \mathbf{x}_i^2 \frac{\mathrm{d}\Gamma_i}{\mathrm{d}t} = \nu \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ji}\Gamma_j \mathbf{x}_i^2 =$$

$$\nu \sum_{j=1}^{N} \Gamma_j \Bigg(\underbrace{\sum_{i=1}^{N} f_{ji}\mathbf{r}_{ji}^2}_{=4} + 2\mathbf{x}_j^x \underbrace{\sum_{i=1}^{N} f_{ji}\mathbf{x}_i^x}_{=0} + 2\mathbf{x}_j^y \underbrace{\sum_{i=1}^{N} f_{ji}\mathbf{x}_i^y}_{=0} - \mathbf{x}_j^2 \underbrace{\sum_{i=1}^{N} f_{ji}}_{=0} \Bigg) =$$

$$4\nu \sum_{i=1}^{N} \Gamma_i = 4\nu I_0. \quad (3.53)$$

□

Due to the non-linear coupling of Γ_i and $\mathbf{x}_i$ in $\mathbf{I}_1$ and I_2, these quantities are generally not exactly conserved when the system of ODEs (3.43) is discretised using Euler's method. Here, one can only verify I_0 to be conserved exactly. Our experiments at the end of this chapter, however, show that all of these properties are conserved very well in practice.

We lastly give a brief remark about the conservation of kinetic energy. In the whole-space case the velocity usually does not decay fast enough for its classical definition $E = \frac{1}{2}\int_{\mathbb{R}^2} |\mathbf{u}|^2 \,\mathrm{d}x$ to be well defined, even if $\mathbf{u}_\infty = 0$. An alternative definition which coincides with the classical one whenever it is well defined is given by $E = \frac{1}{2}\int_{\mathbb{R}^2} \omega\psi \,\mathrm{d}\mathbf{x}$, where ψ is the stream function from the Helmholtz decomposition. In both cases one can see that the energy is related to a double integral over a square of the vorticity, which is not reflected by the moment conditions. We thus cannot expect the scheme to conserve kinetic energy exactly.

3.4.4 Reduced Operator

In this section we describe a simple technique that may be used to reduce the number of particles that need to be considered for diffusion. If we apply the method as described above, also particles that carry negligible amounts of circulation are redistributed. For this reason, Shankar and van Dommelen [64] suggest to only diffuse particles carrying more circulation than a prescribed threshold. In their work, they set this threshold to the machine epsilon for single-precision floating-point arithmetic, i. e., round about $5.96 \cdot 10^{-8}$.

While choosing a threshold near machine accuracy does produce accurate results, this remains an arbitrary choice. It would be preferable to have an adaptive, problem dependent threshold that guarantees accuracy of the resulting discretisation. Our analysis of the error may be extended to allow for the exclusion of particles from diffusion. To this end, let $\mathcal{I} \subset \{1, \ldots, N\}$ denote a set of particles that are not diffused and let $\tilde{\Delta}_h$ refe to the corresponding 'reduced' approximation of the Laplacian:

$$\tilde{\Delta}_h \omega_h := \sum_{i \notin \mathcal{I}} \sum_{j=1}^{N} \Gamma_i f_{ij} \delta(\mathbf{x} - \mathbf{x}_j). \quad (3.54)$$

The error of this reduced operator can be analysed with the help of the techniques introduced above.

THEOREM 3.5. *The reduced operator $\tilde{\Delta}_h$ satisfies the following error bound:*

$$\|(\Delta_h - \tilde{\Delta}_h)\omega_h\|_{W^{-(m+2),1}} \leq C(rh)^{-2}\|(\Gamma_i)_{i\in\mathcal{I}}\|_{l^1}. \tag{3.55}$$

Proof. Let $\varphi \in W^{m+2,\infty}$ be arbitrary but fixed. We then have with Hölder's inequality:

$$\begin{aligned} |\langle(\Delta_h - \tilde{\Delta}_h)\omega_h, \varphi\rangle| &= |\sum_{i\in\mathcal{I}}\sum_{j=1}^{N}\Gamma_i f_{ij}\varphi(\mathbf{x}_j)| \\ &\leq \|(\Gamma_i)_{i\in\mathcal{I}}\|_{l^1}\|(\sum_{j=1}^{N} f_{ij}\varphi(\mathbf{x}_j))_{i\in\mathcal{I}}\|_{l^\infty}. \end{aligned} \tag{3.56}$$

Applying the triangle-inequality, Lemma 3.1 and the Sobolev embedding yields the result. □

It makes sense to require the magnitude of this additional source error to be of the same order as of the error of the full scheme. Introducing a new user defined constant C_{diff} we thus require:

$$\|(\Gamma_i)_{i\in\mathcal{I}}\|_{l^1} \leq C_{\text{diff}}h^{m+2}\|(\Gamma_i)_i\|_{l^1}. \tag{3.57}$$

To minimise the number of diffused particles we can then employ the following greedy strategy. First, the particles are sorted in ascending order by the magnitude of their circulation. One then continues adding particles to $\mathcal{I}$ from the beginning of this list, until the bound (3.57) is reached. Using the same methods as above, it is easily verified that the thereby defined reduced operator does conserve circulation and linear momentum, however, it does *not* conserve angular momentum.

3.4.5 Limitations

We want to finish this section by some remarks on the limitations of the method. First, unlike claimed by Shankar and van Dommelen [64] the vorticity redistribution method *does not* extend to arbitrary orders of accuracy. The consistency and stability proofs rely on the fact that the stencils are non-negative. While other stable stencils might and probably do exist, we are not aware of any stability proof. As mentioned by Seibold [66, 67], any third or higher order method needs to fulfil the moment conditions for $|\alpha| = 4$. A simple linear combination of these equations yields:

$$\sum_{j\neq i} f_{ij}|\mathbf{r}_{ij}|^4 = 0, \tag{3.58}$$

which due to the non-negativity constraint can only be fulfilled for $f_{ij} \equiv 0$. The zero stencil, however, is inconsistent with the moment conditions for $|\alpha| = 2$. On the other hand, similar restrictions apply to the stability proof for the method of particle strength exchange (PSE): it assumes a positive kernel function, which equally limits the method to second order accuracy [48].

Secondly, we point out that the matrix F discontinuously depends on the particle positions: as they move around, they may enter and leave each other's neighbourhoods, allowing for jumps between zero and non-zero in the corresponding entries f_{ij}. In fact, in general, the solution to the moment equations is not even unique. It is thus hard to analyse the effect of higher-order time-stepping schemes on the method's accuracy.

These restrictions may be seen as a consequence of the lack of structure in particle approximations. The Laplacian is an elliptic, symmetric operator, which leads to positive definite matrices in grid-based schemes. These additional structures allow for stability proofs for higher order schemes. In particle field approximations this symmetry can usually not be maintained.

3.5 Implementation

It has been claimed that the VRM is a slow algorithm, especially when compared to the PSE scheme, e. g., by Cottet and Koumoutsakos [48]. On the other hand, Schrader et al. [65] report that their DC-PSE method also takes up as much as 90% of total CPU time, and compare its computational speed with that of the VRM. We believe that the computational cost associated with the VRM has been greatly overestimated; mostly due to implementation issues. In this section we discuss some of these issues and illustrate a heuristic which can further speed up the method significantly. In our final implementation the velocity computation took about three times longer than the evaluation of the discrete Laplacian.

3.5.1 Solution of the Moment Equations

Equation (3.15) is a classical 'phase I problem' of the Simplex algorithm for linear programming problems. When we use an insertion scheme such as the one described in section 3.5.2 this system is underdetermined, with a fixed, small number of rows $n = 5$ or $n = 9$, corresponding to the number of moment conditions, and a variable number of columns, corresponding to the size of the neighbourhood $\mathcal{N}_i \setminus \{i\}$.

The theory of simplex algorithms is to vast to be treated in detail here, such that we can only give a some key remarks and refer to the literature, e. g., Fletcher's book [72], for further details. Assuming that the moment conditions do have a solution, phase I of the algorithm always returns one with n non-zero entries corresponding to a certain subset of particles in the neighbourhood. These non-zero entries are called basic variables. Setting the fractions f_{ij} for the remaining particles to zero, the solution can be obtained by solving an $n \times n$ linear system. The simplex algorithm is a systematic, iterative way of finding a valid set of basic variables. In every iteration of the algorithm an $n \times n$ system consisting of varying sets of columns of V_i needs to be solved, typically by means of an LU decomposition, which is of $\mathcal{O}(n^3)$ complexity [73].

The efficiency of the method thus crucially depends on the number of rows of V_i, which should be kept as small as possible. Shankar and van Dommelen [64] use a different linear programming problem, aiming to minimise the maximum norm of the solution. By doing so, they solve a problem involving $4n$ rows, effectively

making each iteration 64 times more expensive. One should thus keep the original formulation (3.15). Furthermore, optimising the solution with respect to some target value forces to algorithm to enter phase II, which further increases its cost without improving the method's order of convergence. One might try to optimise the error constant by choosing an optimisation criterion that favours close particles. However, in regard of the later introduced heuristic of small neighbourhoods in section 3.5.3, it is not immediately clear if this additional optimisation step is cheaper than choosing smaller values of h.

Note that the two possible values of n are very small and fixed. An efficient implementation should thus make use of this fact: all loops of the LU decomposition can be unrolled, enabling compilers to perform aggressive optimisations. The LAPACK routines, on the other hand, were optimised for larger problems with dynamic, varying sizes [74].

There are several approaches to avoid a from-scratch computation of the LU decomposition in every iteration of the method. Updating LU decompositions instead of recomputing them, however, typically is only effective for larger values of n: the Fletcher–Matthews update, for example, is reported to be effective for $n > 10$ [75].

Note that the matrix V_i is fully populated and—as the number of neighbours is typically limited—of small to moderate size. On the other hand, most available implementations of the Simplex algorithm as well as a substantial part of the available literature focus on large-scale, sparse problems. In other words, they are optimised for the opposite case and thus cannot deliver good performance for our problem. Implementing an efficient, dense simplex method is essential for the overall performance of the VRM. As this task is not straight forward, some authors, e. g., Lakkis and Ghoniem [76], prefer to solve the non-negative least-squares problem instead:

$$\min_{\mathsf{f}_i \geq 0} |\mathsf{V}_i \mathsf{f}_i - \mathsf{b}_i|^2, \tag{3.59}$$

where $|\cdot|$ refers to the Euclidean norm. This problem can be solved using the algorithm due to Lawson and Hanson [77], which solves an *unconstrained* least-squares problem in each iteration. However, the size of this unconstrained problem varies in every iteration, making it harder to unroll loops a priori. Additionally, these problems are typically solved using QR or LQ decompositions, which are more expensive than the LU decomposition. We therefore do not further pursue this approach.

3.5.2 Insertion of New Particles

In order to ensure that non-negative stencils exist, particles need to have sufficiently many neighbours which also need to fulfil certain geometric conditions. Seibold [66, 67] gives the exact conditions for the first order case $m = 1$ as well as the following sufficient condition: seen from the centre of the neighbourhood, the angle between two adjacent particles may be no more than 45°. Assuming a given maximum hole-size in the particle cloud, he also gives a sufficient upper bound Rh for the neighbourhood size. These conditions could in principle be implemented in a VRM scheme, resulting in a strong guarantee that positive stencils always exist.

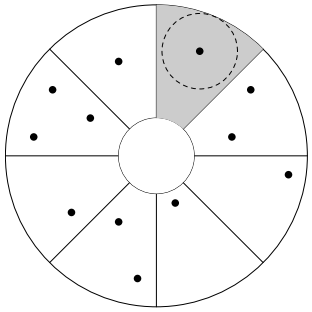

Figure 3.1: Illustration of a particle neighbourhood and the insertion strategy. Each of the eight segments except for the shaded one contained at least one particle. In the shaded segment a new particle is inserted on the centre line at radial position $1.5h$. No particle can be closer than $0.5h$ to the newly inserted particle: the circle of that radius is indicated using a dashed line and is completely included in the previously empty segment.

However, as he points out, these conditions are often too strict. We thus pursue a different approach. Instead of directly checking the angles between each pair of adjacent particles, we subdivide the neighbourhood into eight segments of 45° each, as illustrated in figure 3.1. In order to avoid wasting computational resources, we do not want to insert new particles that would violate the lower bound in (3.10) for any other particle. However, we also want to avoid small values of r, to prevent the time-step constraint (3.42) from becoming too strict. As a compromise we choose $r = \frac{1}{2}$ and $R = 2$ and apply the following insertion strategy: if any neighbourhood segment contains no particles, a new particle is inserted on the segment's centre line at radial position $1.5h$. As illustrated in figure 3.1, this ensures that the newly inserted particle does not violate any other particle's lower bound on its neighbourhood.

This insertion strategy ensures that particles are at most spaced $2h$ apart. According to theorem 6.11 of Seibold's thesis, choosing the upper bound of the neighbourhood size as $R \geq 5.23$ then guarantees the existence of positive stencils. However, in our numerical experiments, such a large choice was not necessary and all computations worked well with $R = 2$. Experiments conducted with a slightly rotated reference frame indicated that the results of this strategy do not significantly depend on the coordinate system used.

Unlike claimed by Cottet and Koumoutsakos [48], insertion of empty particles is different from remeshing: it leaves the vorticity field ω_h unchanged, thereby introducing no error and it does not rearrange existing particles. For this reason the VRM is a truly mesh-free method.

3.5.3 Small Neighbourhoods

As pointed out in section 3.5.1, the simplex method systematically determines a subset of particles leading to a non-negative solution of the moment equations. One

can consequently lower iteration counts by reducing the number of particles in the neighbourhood. In most cases a non-negative solution exists if there is just one particle in every 45°-segment of the neighbourhood. This leads us to the following approach: for every particle neighbourhood, choose the closest particle of each segment. We call the resulting subset the *small neighbourhood*. We then apply the simplex method to this small neighbourhood. By choosing the segments' closest particles, we aim to locally reduce R, thereby minimising the error constant. Only if no non-negative solution was found, we retry with the complete neighbourhood. In our numerical examples, depending on h, this only happened in a negligible (less than a hundred) number of cases.

This approach has the advantage that *all* matrices and vectors involved in the simplex algorithm can be statically allocated, avoiding the overhead of dynamic memory allocation and further enabling the compiler to unroll more loops. In our experiments in section 3.6.2, the use of these small neighbourhoods instead of the complete ones lead to a threefold speed-up.

Note that after the assembly of the Vandermonde matrices V_i, this approach leads to a set of completely decoupled, small problems of fixed size. We thus have an *embarrassingly parallel* problem, making it ideally suited for computations on many-core processors, such as GPUs or the Intel Xeon Phi.

3.6 Numerical Experiments

As Shankar and van Dommelen point out in their work [64], the Lamb–Oseen flow is an ideal test-case for vortex particle methods: its initial condition is a single Dirac delta distribution:

$$\omega(0, \mathbf{x}) = \Gamma\delta(\mathbf{x}), \tag{3.60}$$

and can thus be exactly represented in a vortex particle method. The analytic solution is infinitely smooth and valid for the heat-equation (3.31) as well as the vorticity equation (2.2):

$$\omega(t, \mathbf{x}) = \frac{\Gamma}{4\pi\nu t} e^{-\frac{|\mathbf{x}|^2}{4\nu t}} \qquad (t > 0). \tag{3.61}$$

The corresponding velocity field is given by:

$$\mathbf{u}(t, \mathbf{x}) = \frac{\Gamma}{2\pi|\mathbf{x}|}\left(1 - \exp\left(-\frac{|\mathbf{x}|^2}{4\nu t}\right)\right)\widehat{\boldsymbol{\varphi}}, \tag{3.62}$$

where $\widehat{\boldsymbol{\varphi}}$ refers to the unit vector in circumferential direction at position $\mathbf{x}$. In the following, we will describe several numerical experiments carried out on this flow. Mimicking Shankar and van Dommelen's case of $\mathrm{Re} = 50$, we chose $m = 1$, $\Gamma = 2\pi$, $C_{\mathrm{diff}} = 1$, $\nu = \frac{1}{50}$. We choose higher resolutions, however, and stop time-integration at $t = 1$.

3.6.1 Convergence with respect to h

We consider the cases with and without convection, corresponding to the Navier–Stokes equation and the heat equation, respectively. In the case of the heat equation, we use Euler's method to advance the solution in time and choose a fixed time step:

$$\Delta t = \frac{1}{8}\frac{(rh)^2}{4\nu}. \tag{3.63}$$

As mentioned previously, in vortex methods in unbounded or periodic flows it is common practice to smooth the vorticity field with a radial blob function ζ_σ of blob-width σ. This corresponds to replacing the singular Biot–Savart kernel $\mathbf{K}$ with a regularised one $\mathbf{K}_\sigma$. We use the following second order kernel obtained after Gaussian smoothing:

$$\mathbf{K}_\sigma(\mathbf{x}) = -\frac{(y,-x)^\top}{2\pi|\mathbf{x}|^2}\left(1-\exp\left(-\left|\frac{\mathbf{x}}{\sigma}\right|^2\right)\right), \qquad \mathbf{x} = (x,y)^\top.$$

Our particle insertion strategy guarantees that particles are at most spaced $2h$ apart. Here we follow the common practice and choose σ proportional to h; to ensure sufficient overlap we choose $\sigma = 3h$. A fast multipole method (FMM) similar to that of Dehnen [44] of order $p = 16$ and multipole acceptance criterion $\theta \leq 0.8$ is used to speed up the velocity computation.

Practical experience has shown that higher order time-stepping methods are required to accurately maintain linear and angular momentum in the case of enabled convection. Like Shankar and van Dommelen, we choose the classical Runge–Kutta method (RK4) in this case. In order to resolve particle movement accurately, the time-step is adaptively chosen as the minimum of (3.63) and the following CFL-type condition:

$$\Delta t \leq \frac{1}{8}\min_{i=1,\dots,N}\frac{h}{|\mathbf{u}_i|}. \tag{3.64}$$

We want to stress that this second bound is not required to ensure stability: experiments without this restriction showed no instabilities and gave reasonable results, however, the errors in linear and angular momentum were larger.

As it is difficult to compute Sobolev-norm $\|\cdot\|_{W^{-m,1}}$ explicitly, we try to approximate the L^2-error of the corresponding velocity. As the system contains infinite energy, we need to limit the area of integration. We define our computational domain as $\Omega = [-1.5, 1.5]^2$, as all particles were contained within this region. By means of numerical quadrature we then evaluate:

$$e_{\mathbf{u}} = \frac{\|\mathbf{u}-\mathbf{u}_\sigma\|_{L^2(\Omega)}}{\|\mathbf{u}\|_{L^2(\Omega)}}, \tag{3.65}$$

where $\mathbf{u}_\sigma$ stands for the velocity field which is obtained from the particle approximation using the smoothed kernel $\mathbf{K}_\sigma$.

Figure 3.2 shows the observed error estimates for various values of h. Even though the expected convergence rate was $m = 1$, we actually observe second order convergence behaviour. This is similar to the observations by Seibold, who explains this using a symmetry argument: the classical five-point finite-difference stencil achieves second order accuracy due to the symmetry of the particle locations. However, the insertion strategy and the definition of the particle neighbourhoods preclude extreme cases of asymmetry, which might result in the observed second order convergence. Seibold, however, does not exclude particles according to equation (3.57). It is thus comes as a surprise that even the reduced operator exhibits this behaviour. As both curves form a nearly straight line and essentially coincide, we suspect that the smoothing error dominates for this choice of parameters.

Figure 3.3 shows the number of particles in the final time-step of the computation. It increases approximately as $\mathcal{O}(h^{-2})$, as one would expect in a grid-based computation. This again is surprising, as bound (3.57) gets stricter for decreasing h. Due to the convection in the Navier–Stokes case, more particles need to be inserted as they move around. In our simulation, this caused an increase in the number of particles of a nearly constant factor 1.6.

As shown in section 3.4.4, the reduced operator conserves circulation and linear momentum exactly. In the case of the heat equation this remains the case when a time-stepping scheme is applied: the error in I_0 and $\mathbf{I}_1$ was of the order of the machine accuracy. For the Navier–Stokes equation this is only true for the circulation. For all choices of h the error in linear momentum varied between $\mathcal{O}(10^{-6})$ and $\mathcal{O}(10^{-7})$. We believe this to be a result of the limited accuracy of the FMM code that was used for the velocity computation and the error introduced by the Runge–Kutta method. Figure 3.4 shows the error in angular momentum I_2. The values for the heat equation decrease at a rate of $\mathcal{O}(h^3)$, similar to the bound (3.57). In the convective case the error decays somewhat faster, in a less clear-cut manner. We believe this to be a result of the increased number of particles. We thus conclude that for the chosen values of h, the error in angular momentum induced by using the reduced operator (3.54) dominates that of the FMM and the time-stepping scheme.

Figure 3.5 shows the velocity at the particle locations for $h = 0.04$ at $t = 1$ with enabled convection. Despite the asymmetry in the particle locations caused by the convection, one can see that the velocity field remains quite symmetric. The reduced operator prevents the creation of particles that would carry insignificant amount of circulation. For this reason, the particle cloud takes the shape of a circle around the origin: vorticity decays exponentially with the distance to the origin. At $\sigma = 3h = 0.12$ the resolution is not high enough to accurately represent steep velocity gradient at the centre of the flow. However, due to the good conservation properties, we obtain a qualitatively good solution already at this under-resolved computation.

3.6.2 Computational Speed

In order to assess the speed of the method, we measured the time needed to evaluate the velocity and the Laplacian for $h = 0.01$. For the Laplacian, we compared the

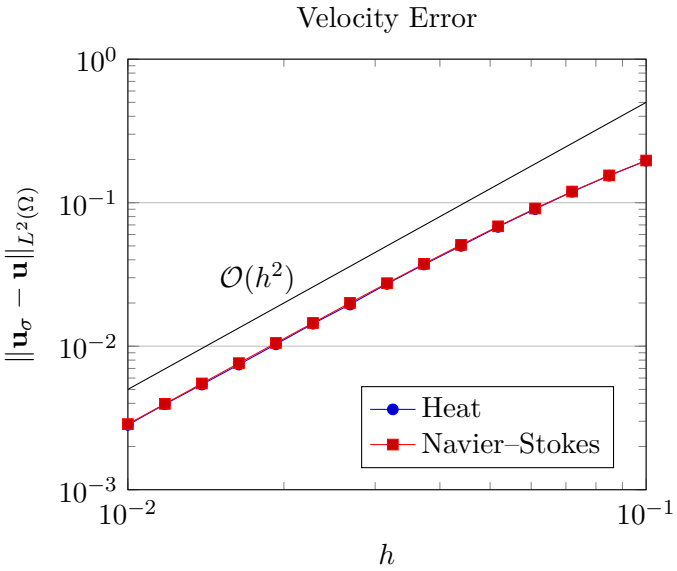

Figure 3.2: Error estimates for the heat and Navier–Stokes equations for varying values of h at $t = 1$. Their values essentially coincide and exhibit an $\mathcal{O}(h^2)$ convergence behaviour.

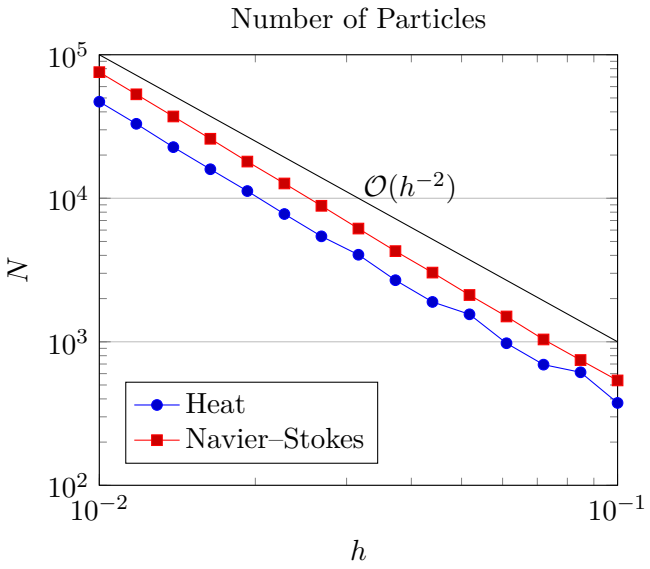

Figure 3.3: The number of particles in the final step of the computation for the heat and Navier–Stokes equations. The curves show a particle growth that scales as $\mathcal{O}(h^{-2})$, despite the fact that equation (3.57) is getting stricter for decreasing mesh-sizes. The ratio between the two curves' values remains approximately fixed at around 1.6.

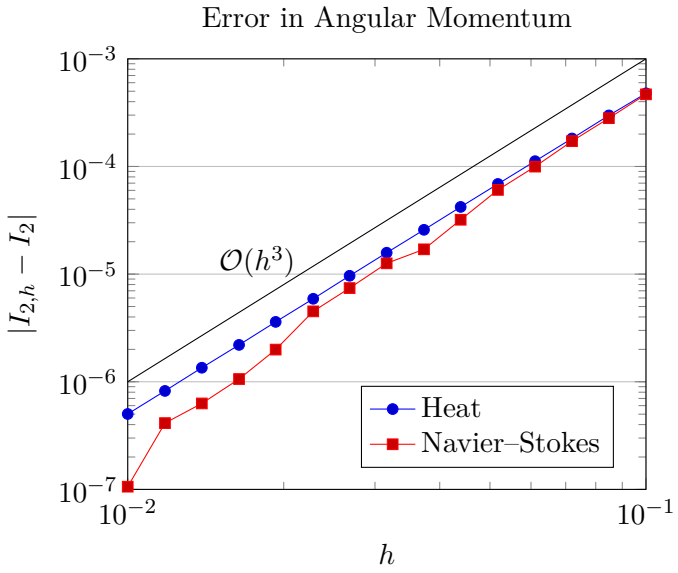

Figure 3.4: Error in angular momentum at the final time-step for the heat and Navier–Stokes equations. The error decays at a rate of $\mathcal{O}(h^3)$, the same exponent as in condition (3.57). In case of the Navier–Stokes equations, the error decreases even faster, in a less clear-cut manner.

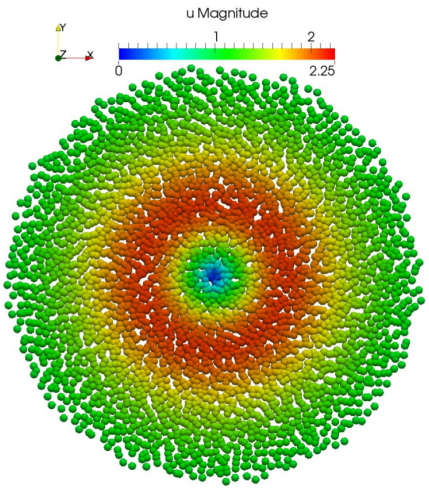

Figure 3.5: Plot of the smoothed velocity $\mathbf{u}_\sigma$ at the particle locations for $h = 0.04$ at $t = 1$. Despite the asymmetric particle distribution, caused by the convection, the velocity field remains very symmetric. The particle cloud takes the shape of a circle. Even in this under-resolved case, the method yields qualitatively good results.

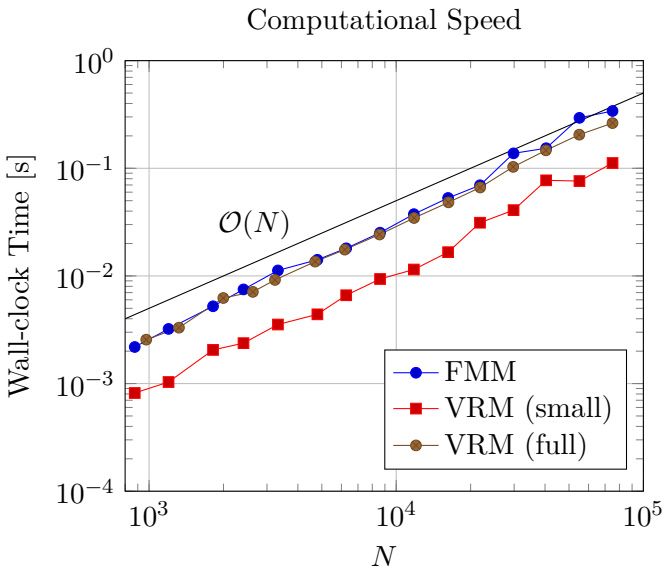

Figure 3.6: Required CPU time for the VRM with the full and small neighbourhoods in comparison to the FMM. The computations were performed on an Intel Xeon E5-1650v3, a six-core processor running at 3.5 GHz. The line corresponding to the FMM is jagged due to the task-based parallelism used in the implementation. The VRM computation can be greatly accelerated using small neighbourhoods, it is then about three times faster than the corresponding velocity computation.

performance of two codes: the first code uses LAPACK to decompose the arising linear systems in each simplex iteration and takes the complete particle neighbourhood into account. The second code uses small neighbourhoods as described in section 3.5.3 and an implementation using completely unrolled loops in the LU decomposition. The code was parallelised using OpenMP, where task based parallelism was used for the FMM.

In an inviscid vortex method the velocity evaluation takes up almost all of the computational time, even when fast algorithms like the FMM are used. Many of the viscous schemes that rely on frequent remeshing, like the method of particle strength exchange (PSE), are so efficient, that the time needed for them is also negligible compared to the velocity evaluation. On the other hand, Schrader et al. [65] report that their DC-PSE method takes up as much as 90% of the total CPU time, and compare its computational speed with that of the VRM. To illustrate that this is not necessarily true, and to show that the VRM is a computationally feasible scheme, we compare the time needed for it with that of the FMM.

Figure 3.6 shows the required time for each computation depending on the number of particles involved. One can see that all computations scale linearly with N, however, with different constant factors. The code using small neighbourhoods performs about three times faster than the corresponding code using the complete ones. This clearly

highlights the benefit of trying small neighbourhoods first. It also performs about three times as fast as the corresponding FMM code. Further measurements showed that, in the case of small neighbourhoods, only about one third of the time was used for the actual simplex solver, while the remaining time was spent finding neighbourhoods and inserting new particles. A hash based algorithm was used for this, causing the resulting curve to be jagged due to caching effects.

Note that these numbers cannot be directly compared to those reported by Shankar and van Dommelen: they compare a single VRM computation to that of a convective step performed using the Runge–Kutta method, i. e., involving four velocity computations. In this setting, their VRM computation takes about 25% longer than the convective step, i. e., five times longer than a single velocity evaluation. In comparison to the respective FMM codes, our VRM computation thus is about 15 times faster.

3.7 Conclusion and Outlook

We have introduced a splitting-free variant of the vorticity redistribution method (VRM). Using the new concept of small-neighbourhoods, its speed compared to the original method can be greatly accelerated and typically is below that of the corresponding velocity computation. Equation (3.57) allows us to efficiently and consistently reduce the number of diffused particles. We have illustrated that the method can be implemented efficiently and that previous claims on the slow speed of the VRM are probably due to implementation issues. The large number of small, independent, fixed-size problems involved makes it an ideal candidate for parallelisation on coprocessors such as GPUs or the Intel Xeon Phi. We conclude this text with a few possible extensions on the method.

In light of the quadratic time-step bound (3.42), an interesting topic for future research might be the application of implicit time-stepping schemes in periodic flows. As the convective part of the equations is non-stiff, this seems to be an ideal use-case for IMEX multistep schemes [78]. After having convected the particles, F could then be readily assembled, leading to a linear system. As Seibold discusses in his work [69], due to the positivity and sparsity of the stencils, such systems can effectively be solved using algebraic multigrid methods.

The definition of a particle's neighbourhood in equation (3.10) excludes particles that are too close to that particle. In order to save computational resources, it may thus be desirable to remove particles in areas where they get too close to one another. As described in section 3.2 instead of approximating the Laplacian, one can apply the same methodology to approximate the identity operator using a particle's neighbours. This way, a particle can be redistributed to its neighbours and subsequently be removed. Lakkis and Ghoniem [76] successfully applied a similar procedure and reported a significant reduction in the number of particles.

Chapter 4

Particle Regularisation

In this chapter we introduce a new method to regularise particle fields in general bounded domains $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$. We begin this chapter by recapitulating the particle regularisation problem which lies at the core of all modern vortex methods. We then describe some commonly used strategies to tackle this problem and their drawbacks. Our new approach is based on two main ingredients: a certain class of smooth approximation spaces and a stabilised variational formulation. We will carefully define these spaces and analyse the variational formulation in the subsequent sections. Our analysis will in particular show that our approach is optimal, in the sense that the regularised vorticity fulfils the same asymptotic error bound as the given particle approximation. Practical experiments will confirm the analysis and illustrate the practicality of the approach.

4.1 Introduction

In order to understand the problem of particle regularisation and its fundamental importance to vortex methods, let us reconsider the simple case of an inviscid, two-dimensional flow in a bounded domain $\Omega = D \subset \mathbb{R}^2$. In this case the transport equation for the scalar vorticity ω reads:

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega = 0 \qquad \text{in } \Omega. \tag{4.1}$$

Let us for the moment consider that the velocity field $\mathbf{u}$ would be known and would satisfy a no-through condition on the domain's boundary $\partial\Omega$. In vortex methods one then discretises the vorticity field with *particles*:

$$\omega(t = 0) \approx \omega_h(0) = \sum_{i=1}^{N} \Gamma_i \delta(\mathbf{x} - \mathbf{x}_i(0)), \tag{4.2}$$

where Γ_i and $\mathbf{x}_i$ denote the circulation and position of particle i, and δ is the Dirac delta function. For such particle discretisations one typically has error bounds of the form $\|\omega - \omega_h\|_{W^{-(m+1),2}(\Omega)} \leq Ch^{m+1}\|\omega\|_{W^{m+1,2}(\Omega)}$, $m \geq 1$, as for example shown in theorem 2.4. The reason for choosing this particular form of discretisation lies in the fact that for such particle fields ω_h an analytic solution of equation (4.1) is available: one simply modifies the particles' positions according to $\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{u}(t, \mathbf{x}_i(t))$, $i = 1, \ldots, N$.

This is the exact solution, thus no additional error is introduced. In particular, at any time $t \geq 0$, the error can be bounded by the initialisation error.

In practice, however, the velocity field is of course not known and needs to be retrieved from the vorticity as described in section 2.2. Let us for simplicity assume that the velocity fulfils $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$, such that we can ignore the boundary integral term $\mathbf{u}_{\partial D}$. In this case the velocity is simply given by the Biot–Savart law as described in subsection 2.2.1:

$$\mathbf{u} = \mathbf{K} \star \omega, \ \mathbf{K}(\mathbf{x}) = \frac{1}{2\pi} \frac{(-x_2, x_1)^\top}{|\mathbf{x}|^2}, \tag{4.3}$$

where $\star$ denotes convolution. As explained in subsection 2.2.4, we have the following classical estimate due to Calderón and Zygmund [53]:

$$\|\mathbf{K} \star \omega\|_{W^{1,2}(\Omega)} \leq C \|\omega\|_{L^2(\Omega)}, \tag{4.4}$$

where here and throughout this text the symbol C refers to a generic positive constant which is independent of the functions involved. The problem is that the particle approximation $\omega_h \notin L^2(\Omega)$ is not smooth enough to apply this estimate; applying the Biot–Savart law to the particle field directly yields a singular velocity field. The question we try to answer in this chapter is how to obtain an accurate, smooth approximation $\omega_\sigma \in L^2(\Omega)$ from the particle field ω_h, where σ refers to a smoothing length, which will be defined precisely later. This is the particle regularisation problem.

Closing our discussion of the introductory example, once a suitable smooth approximation ω_σ has been obtained, one sets $\mathbf{u}_\sigma := \mathbf{K} \star \omega_\sigma$ and evolves the particle locations according to $\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{u}_\sigma(t, \mathbf{x}_i(t))$, instead. One can then show that the resulting method converges, and due to its natural treatment of convection is essentially free of artificial viscosity—one of the main draw backs of conventional grid based methods [48, Section 2.6]. All extensions to this scheme, e. g., three-dimensional or viscous vortex methods also rely on the availability of such a regularisation procedure.

The most common approach to the regularisation problem is to mollify the particle field with a certain, radially symmetric *blob-function* ζ_σ: $\omega_\sigma := \omega_h \star \zeta_\sigma$, where σ denotes the radius of the blob's core [48, Section 2.3]. Many commonly used blob-functions have infinite support, effectively extending ω_σ from Ω to $\mathbb{R}^d$ and blurring the domain's boundaries. Additionally, instead of evaluating the Biot–Savart integral as $\mathbf{K} \star \omega_\sigma$, one usually first computes $\mathbf{K}_\sigma := \mathbf{K} \star \zeta_\sigma$ explicitly and then evaluates the equivalent expression $\mathbf{u}_\sigma = \mathbf{K}_\sigma \star \omega_h$. This complicates the application of fast summation algorithms such as the fast multipole method, which rely on series expansions of the kernel function. Such expansions are available for $\mathbf{K}$, but this is typically not the case for the smoothed kernel $\mathbf{K}_\sigma$.

There are approaches to use blob-functions with varying shapes near boundaries [76, 79] or to use image particles outside of the domain [48, Section 4.5.2]. These approaches assume that the boundaries are flat and usually fail in the presence of sharp corners or kinks. Another approach is to interpret the particles' circulations as weighted function values $\Gamma_i = w_i \omega(\mathbf{x}_i)$, where the w_i are weights from an underlying quadrature rule. While this is strictly speaking only the case during the initialisation stage, the approach is then to create a triangulation of the domain using the particles' positions

as grid nodes and to use these values to construct a piecewise linear approximation [80]. This requires a mesh to be regenerated at every time-step, which is problematic as the particle field gets distorted over time. In Vortex-in-Cell (VIC) schemes one uses interpolation formula to obtain a grid-based approximation of the vorticity field. In the vicinity of boundaries these formulae need to be specifically adapted to the particular geometry at hand and cannot be used for general domains [81]. In summary it can be said the regularisation problem causes significant difficulties, and as Cottet and Koumoutsakos point out in the introduction of their book [48]: 'To our knowledge there is no completely satisfactory solution for general geometries, in particular because of the need to regularize vortices near the boundary.' Marichal, Chatelain, and Winckelmans come to a similar conclusion in their recent review of regularisation schemes [82]: 'None of the schemes above truly succeeds in the generation of accurate particle—or grid—values around boundaries of arbitrary geometry.' The implications of this observation are severe: without an adequate regularisation procedure, vortex methods cannot be applied to flows in bounded domains.

In this chapter we try to address this problem with the help of a finite element formulation. The non-smooth $W^{-(m+1),2}$-nature of particle field approximations forces us to use shape functions that are globally at least $W^{m+1,2}$-smooth, which is not the case for the classical, piecewise linear elements. The partition of unity finite element method (PUFEM) by Melenk and Babuška is a generalisation of the classical finite element method (FEM), which can be used to obtain such smooth spaces. Even though already mentioned in their introductory paper [83], there seems to have been little research in this direction. Duarte et al. [84] describe an approach which only works for certain triangulations in two dimensions.

The generation of globally smooth shape functions on general meshes in higher dimensions is a well-known, hard problem. We instead consider simple Cartesian grids, on which the construction of smooth shape functions is easier. We then apply a fictitious domain approach to deal with general geometries. This typically results in instabilities in the cut elements. Under the name *ghost penalty* Burman [85] presented an effective and accurate stabilisation strategy for this problem, which has for example been successfully applied to several other flow problems with cut elements [86, 87, 88, 89]. We use a similar, higher-order approach inspired by Cattaneo et al. [90] as well as Burman and Fernández [91] to achieve accuracy and stability of the resulting discretisation.

The rest of this chapter is structured as follows. In section 4.2 we define and construct smooth PUFEM spaces and analyse some of their important properties. In section 4.3 we introduce a stabilised variational formulation and prove its stability and convergence. The regularisation problem is then treated as a perturbation to this variational formulation. Similar to the approach with blob-functions as mentioned above, the resulting error can be split into regularisation and quadrature error parts which need to be carefully balanced. The analysis will show that the smoothing parameter σ should be taken proportional to the square-root of the particle-spacing h and that this choice is in a certain sense optimal. In section 4.4 we perform numerical experiments, confirming our analysis. As a consequence of the quadratic relation between h and σ, the computation of the velocity field only has a computational

complexity of $\mathcal{O}(h^{-\frac{d}{2}})$, enabling the use of particle numbers on desktop workstations which were previously only possible on super computers. We finish this chapter with some concluding remarks and an outlook to possible future extensions.

4.2 Smooth Partition of Unity Finite Element Spaces

In this section we are going to introduce smooth PUFEM spaces, which are necessary for the application of the particle approximation ω_h to be well defined. We investigate their approximation qualities and give an explicit construction for such spaces which can be used in a computer implementation. We conclude this section with inverse estimates, which will be needed in the analysis of the variational formulation in the next section.

4.2.1 Basic Theory

We begin this subsection by defining smooth partitions of unity, similar to Melenk and Babuška's theory [83].

Definition 4.1 (Smooth Partition of Unity). *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and let $\{\Omega_i\}$ be an open cover of Ω satisfying a pointwise overlap condition:*

$$\exists M \in \mathbb{N} : \ \forall x \in \Omega : \ \operatorname{card}\{i \,|\, x \in \Omega_i\} \leq M. \tag{4.5}$$

Let $\{\varphi_i\}$ be a Lipschitz partition of unity subordinate to the cover $\{\Omega_i\}$ satisfying

$$\operatorname{supp} \varphi_i \subset \operatorname{clos} \Omega_i, \tag{4.6}$$

$$\sum_i \varphi_i(x) \equiv 1 \text{ on } \Omega, \tag{4.7}$$

$$|\varphi_i|_{W^{k,\infty}(\mathbb{R}^d)} \leq C(k)(\operatorname{diam} \Omega_i)^{-k} \quad k \in \mathbb{N}_0, \tag{4.8}$$

where the $C(k)$ are positive constants and the symbol $|\cdot|_{W^{k,p}(\mathbb{R}^d)}$ refers to the Sobolev semi-norms:

$$|f|_{W^{k,p}(\mathbb{R}^d)} := \begin{cases} \left(\sum_{|\alpha|=k} \|\partial^\alpha f\|^p_{L^p(\mathbb{R}^d)} \right)^{1/p} & p \in [1, \infty), \\ \max_{|\alpha|=k} \|\partial^\alpha f\|_{L^\infty(\mathbb{R}^d)} & p = \infty. \end{cases} \tag{4.9}$$

Then, $\{\varphi_i\}$ is called a smooth partition of unity subordinate to the cover $\{\Omega_i\}$. The sets Ω_i are called patches.

Using these functions $\{\varphi_i\}$, we can define the spaces for the partition of unity finite element method (PUFEM).

Definition 4.2 (PUFEM Spaces). *Let $\{\varphi_i\}$ be a smooth partition of unity subordinate to the open cover $\{\Omega_i\}$. For $P, k \in \mathbb{N}_0$, $p \in [1, \infty]$ we define polynomial enrichment spaces $V_i^P \subset W^{k,p}(\Omega \cap \Omega_i)$:*

$$V_i^P := \operatorname{span}\{\mathbf{x}^\alpha \,|\, |\alpha| \leq P\}, \tag{4.10}$$

and the PUFEM spaces $V_\sigma^P(\Omega) \subset W^{k,p}(\Omega)$:

$$V_\sigma^P := \operatorname{span}\{\varphi_i v_i \,|\, v_i \in V_i^P\} \tag{4.11}$$

where $\sigma := \max_i \operatorname{diam} \Omega_i$ *refers to the maximum patch diameter.*

Assumption 4.3. We will assume that the shapes of the domain Ω and the patches $\{\Omega_i\}$ are such that we can apply the Bramble–Hilbert lemma [60, Lemma (4.3.8)]. In particular, we will assume that for all $u \in W^{P+1,p}(\Omega \cap \Omega_i)$, $p \in [1,\infty]$, there exists a $v_i \in V_i^P$ such that:

$$|u - v_i|_{W^{k,p}(\Omega\cap\Omega_i)} \leq C\sigma^{P+1-k}|u|_{W^{P+1,p}(\Omega\cap\Omega_i)} \;\forall k \in \mathbb{N}_0,\; k \leq P+1, \tag{4.12}$$

where the constant C is independent of σ and u.

We then have the following estimate, which is a straightforward generalisation of the result of Melenk and Babuška [83, Theorem 2.1].

THEOREM 4.4. *Let* $V_\sigma^P(\Omega)$ *be as in definition 4.2 and let assumption 4.3 be fulfilled. Then for any* $u \in W^{P+1,p}(\Omega)$, $p \in [1,\infty]$, *there exists* $\mathcal{P}u \in V_\sigma^P(\Omega)$ *such that:*

$$|u - \mathcal{P}u|_{W^{k,p}(\Omega)} \leq C\sigma^{P+1-k}|u|_{W^{P+1,p}(\Omega)} \;\forall k \in \mathbb{N}_0,\; k \leq P+1, \tag{4.13}$$

where the constant C *is independent of* σ *and* u.

Proof. Here, we will only consider the case $p \in [1,\infty)$; the proof for the case $p = \infty$ is analogous. With $v_i \in V_i^P$ as in assumption 4.3 we set $\mathcal{P}u := \sum_i \varphi_i v_i$. We may then write for any multi-index α with $|\alpha| = k$:

$$\|\partial^\alpha(u - \mathcal{P}u)\|_{L^p(\Omega)}^p = \|\partial^\alpha \sum_i (u - v_i)\varphi_i\|_{L^p(\Omega)}^p = \|\sum_{\beta\leq\alpha}\sum_i \binom{\alpha}{\beta} \partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\|_{L^p(\Omega)}^p. \tag{4.14}$$

Considering the absolute value of the expanded derivative on the right, we obtain using Hölder's inequality:

$$\left|\sum_{\beta\leq\alpha}\sum_i \binom{\alpha}{\beta} \partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\right|^p \leq C(\alpha,p)\sum_{\beta\leq\alpha}\left|\sum_i \partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\right|^p \tag{4.15}$$

and thus:

$$\|\partial^\alpha(u - \mathcal{P}u)\|_{L^p(\Omega)}^p \leq C\sum_{\beta\leq\alpha} \|\sum_i \partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\|_{L^p(\Omega)}^p. \tag{4.16}$$

Now, using the fact that for every point $x \in \Omega$ there are at most M non-zero terms in the sum over i, we obtain by again using Hölder's inequality:

$$\left|\sum_i \partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\right|^p \leq C(M,p)\sum_i |\partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)|^p \tag{4.17}$$

After inserting this in the previous relation we may exchange the order of summation. Using the fact that $\varphi_i \equiv 0$ outside Ω_i we obtain:

$$\|\partial^\alpha(u - \mathcal{P}u)\|_{L^p(\Omega)}^p \leq C\sum_i\sum_{\beta\leq\alpha} \|\partial^\beta\varphi_i \partial^{\alpha-\beta}(u - v_i)\|_{L^p(\Omega\cap\Omega_i)}^p. \tag{4.18}$$

Applying (4.8), (4.12), and the finite overlap condition (4.5) yields the claim. □

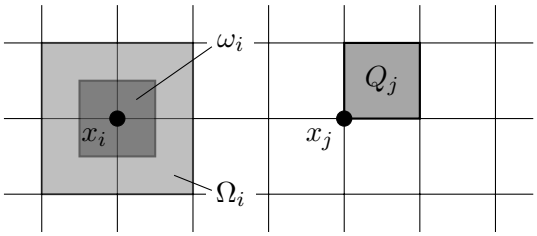

Figure 4.1: An illustration of the Cartesian grid. On the left a grid node x_i together with its associated patch-core ω_i and patch Ω_i. On the right another grid node x_j with its associated element Q_j.

4.2.2 Construction of a Smooth Partition of Unity

In this subsection we are going to construct such a smooth partition of unity using mollification. We will make use of the following two definitions.

Definition 4.5 (Friedrichs' Mollifier). *The function:*

$$\zeta : \mathbb{R} \to [0, K^{-1}], \qquad x \mapsto \begin{cases} 0 & \text{if } |x| \geq \frac{1}{2}, \\ K^{-1} \exp\left(-\frac{1}{1-4x^2}\right) & \text{else,} \end{cases} \tag{4.19}$$
$$K \approx 0.221\,996\,908\,084\,039\,719,$$

is called Friedrichs' mollifier in one-dimensional space. The constant K was obtained numerically, such that $\|\zeta\|_{L^1(\mathbb{R})} = 1$. For spatial dimensions greater than $d = 1$ we define Friedrichs' mollifier using the product:

$$\zeta : \mathbb{R}^d \to [0, K^{-d}], \qquad (x_1, \ldots, x_d) \mapsto \prod_{i=1}^{d} \zeta(x_i),$$

where under a slight abuse of notation, we reused the symbol ζ.

It is well known that $\zeta \in C_0^\infty(\mathbb{R}^d)$, and thus also $\zeta \in W^{k,p}(\mathbb{R}^d),\ k \in \mathbb{N}_0,\ p \in [1, \infty]$. Furthermore, we have $\operatorname{supp} \zeta = [-\frac{1}{2}, \frac{1}{2}]^d$, which leads to the following definition.

Definition 4.6 (Cartesian Grid). *Given $\sigma > 0$, we define Cartesian grid points $\mathbf{x}_i \in \mathbb{R}^d$, $i \in \mathbb{Z}^d$, $\mathbf{x}_i := (i_1\sigma, \ldots, i_d\sigma)$. With each grid point we associate a patch Ω_i and a patch-core ω_i:*

$$\Omega_i := \big((i_1 - 1)\sigma, (i_1 + 1)\sigma\big) \times \ldots \times \big((i_d - 1)\sigma, (i_d + 1)\sigma\big), \tag{4.20}$$
$$\omega_i := \big((i_1 - \tfrac{1}{2})\sigma, (i_1 + \tfrac{1}{2})\sigma\big) \times \ldots \times \big((i_d - \tfrac{1}{2})\sigma, (i_d + \tfrac{1}{2})\sigma\big). \tag{4.21}$$

An illustration of these definitions is given in figure 4.1. It is obvious that the patches $\{\Omega_i\}$ form an open cover of $\mathbb{R}^d$ with M from definition 4.1 being equal to 2^d. The patch-cores $\{\omega_i\}$ are pairwise disjoint and their closures form a (non-open) cover of $\mathbb{R}^d$. Using these definitions, we are now ready to construct smooth partition of unity functions $\{\varphi_i\}, i \in \mathbb{Z}^d$.

LEMMA 4.7. *For a given $\sigma > 0$ and $i \in \mathbb{Z}^d$ let φ_i be the convolution of the characteristic function χ_{ω_i} of the patch-core ω_i with the scaled Friedrichs' mollifier $\zeta_\sigma(\mathbf{x}) := \sigma^{-d}\zeta(\mathbf{x}/\sigma)$:*

$$\varphi_i(\mathbf{x}) := (\chi_{\omega_i} \star \zeta_\sigma)(\mathbf{x}) = \int_{\omega_i} \zeta_\sigma(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{4.22}$$

One then has:

$$\varphi_i \in C_0^\infty(\mathbb{R}^d) \text{ and } \operatorname{supp}\varphi_i = \operatorname{clos}\Omega_i, \tag{4.23}$$

$$\sum_{i\in\mathbb{Z}^d} \varphi_i(\mathbf{x}) \equiv 1 \quad \mathbf{x} \in \mathbb{R}^d, \tag{4.24}$$

$$|\varphi_i|_{W^{k,p}(\mathbb{R}^d)} \leq C(k)\sigma^{d/p-k} \quad k \in \mathbb{N}_0,\ p \in [1,\infty]. \tag{4.25}$$

Proof. The first property directly follows from the classical properties of mollification [68, sections 2.28 and 2.29]. For the second property we immediately obtain:

$$\sum_{i\in\mathbb{Z}^d} \varphi_i(\mathbf{x}) = \sum_{i\in\mathbb{Z}^d} \int_{\omega_i} \zeta_\sigma(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbb{R}^d} \zeta_\sigma(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y} = 1. \tag{4.26}$$

For the last property we obtain with the help of Young's inequality for convolutions for every multi-index α with $|\alpha| = k$:

$$\|\chi_{\omega_i} \star \partial^\alpha\zeta_\sigma\|_{L^p(\mathbb{R}^d)} \leq \|\chi_{\omega_i}\|_{L^p(\mathbb{R}^d)}\|\partial^\alpha\zeta_\sigma\|_{L^1(\mathbb{R}^d)} = \sigma^{d/p-k}\|\partial^\alpha\zeta\|_{L^1(\mathbb{R}^d)}. \tag{4.27}$$

□

Remark 4.8. There is no closed-form expression for the functions $\{\varphi_i\}$ available. However, it is important to notice that we have:

$$\varphi_i(\mathbf{x}) \equiv \hat{\varphi}\left(\frac{\mathbf{x} - \mathbf{x}_i}{\sigma}\right), \quad \hat{\varphi}(\mathbf{x}) := \int_{(-\frac{1}{2},\frac{1}{2})^d} \zeta(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{4.28}$$

Furthermore, $\hat{\varphi}$ inherits the product structure of ζ. In a computer implementation it is thus sufficient to tabulate values for $\hat{\varphi}$ corresponding to the case $d = 1$. We can then efficiently approximate $\hat{\varphi}$ using, e. g., cubic Hermite splines. The graph of this function can be seen in figure 4.2.

4.2.3 Reference Element and Inverse Estimates

In this subsection we illustrate that the smooth partition of unity constructed in subsection 4.2.2 leads to spaces that can be treated in a manner similar to conventional finite element spaces. In particular, we can subdivide $\mathbb{R}^d$ into *elements*:

$$Q_i := (i_1\sigma, (i_1+1)\sigma) \times \ldots \times (i_d\sigma, (i_d+1)\sigma), \qquad i \in \mathbb{Z}^d. \tag{4.29}$$

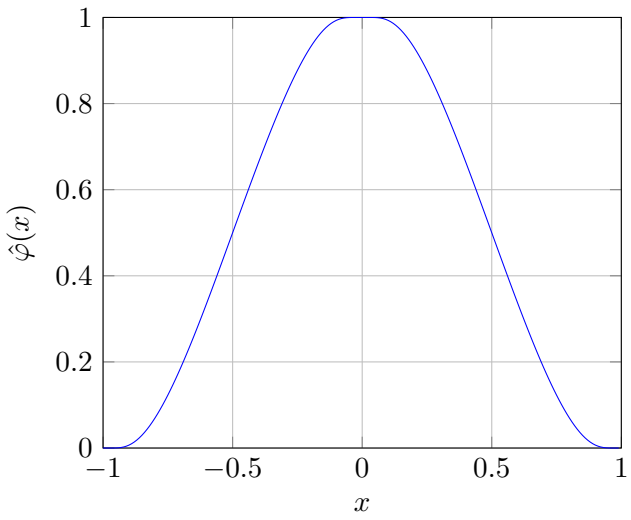

Figure 4.2: An illustration of the one-dimensional partition of unity function $\hat{\varphi}$.

Every Q_i may then be seen as the image of the *reference element* $\hat{Q} := (0,1)^d$ under the transformation $\Phi_i : \hat{Q} \to Q_i,\ \hat{\mathbf{x}} \mapsto \mathbf{x}_i + \sigma\hat{\mathbf{x}}$. In every element Q_i we have a fixed set $\mathcal{J}_i$ of 2^d overlapping patches Ω_j. Introducing:

$$\mathcal{B}_j^P := \Big\{ \underbrace{\left(\frac{\mathbf{x}-\mathbf{x}_j}{\sigma}\right)^{\alpha}}_{=:g_{j,\alpha}(\mathbf{x})} \Big| \, |\alpha| \leq P \Big\}, \qquad j \in \mathbb{Z}^d \tag{4.30}$$

as bases for the enrichment spaces V_j^P, one quickly sees that within each element Q_i the basis functions $g_{j,\alpha}$ can be expressed in terms of mapped reference functions $\hat{g}_{m,\alpha}$:

$$g_{j,\alpha}(\mathbf{x}) = (\hat{g}_{m,\alpha} \circ \Phi_i^{-1})(\mathbf{x}), \qquad \mathbf{x} \in Q_i,\ j \in \mathcal{J}_i, \tag{4.31}$$

where m is the index of the node in the reference element that corresponds to $\mathbf{x}_j$. Due to remark 4.8, the same holds true for the partition of unity functions φ_j. This allows us to infer the following classical result, which follows from a scaling argument and the norm-equivalence of finite-dimensional spaces [60, Lemma (4.5.3)].

Lemma 4.9 (Inverse Estimates). *Let $V_\sigma^P(\Omega)$, $\sigma > 0$, $P \in \mathbb{N}_0$ be defined as above. Then, for any element $Q_i \subset \Omega$ contained in the domain and every $v_\sigma \in V_\sigma^P(\Omega)$ one has:*

$$\|v_\sigma\|_{W^{l,p}(Q_i)} \leq C\sigma^{k-l}\|v_\sigma\|_{W^{k,p}(Q_i)}, \quad p \in [1,\infty],\ k,l \in \mathbb{N}_0,\ k \leq l, \tag{4.32}$$

where the constant C is independent of σ and i.

4.3 Stabilised Variational Formulation

In this section we will introduce a stabilised variational formulation with the aim of mimicking of the $L^2(\Omega)$-orthogonal projector onto $V_\sigma^P(\Omega)$. As the inverse estim-

ates (4.32) are not available for elements Q_i cut by the boundary $\partial\Omega$, we will employ a fictitious domain approach. In order to ensure coercivity of the resulting bilinear form on the entire fictitious domain, we will add a stabilisation term in the cut cells. Once consistency and stability of this formulation have been established, we will model the regularisation process as a perturbation to this variational problem.

4.3.1 Basic Definitions and Conditions

We will restrict ourselves to Hilbert spaces ($p = 2$), due to the rich theoretical framework available for this case. We will assume that the shape of the domain Ω is such that we may apply the Stein extension theorem [68, Section 5.24], i. e., there exists a bounded linear extension operator $\mathcal{E} : W^{k,2}(\Omega) \to W^{k,2}(\mathbb{R}^d)$ for any natural number k. We explicitly wish to include functions that do not vanish on the boundary $\partial\Omega$. For this reason, for any domain $\square \subset \mathbb{R}^d$, we will denote by $W^{-k,2}(\square) := W^{k,2}(\square)'$ the dual space of $W^{k,2}(\square)$. (Opposed to the convention $W^{-k,2}(\square) = W_0^{k,2}(\square)'$).

We will need certain geometrical definitions. To this end, let $\sigma > 0$ be arbitrary but fixed. We define the fictitious domain Ω_σ as the union of all elements that intersect the physical domain Ω. Furthermore we define cut and uncut elements Ω_σ^Γ and Ω_σ°, respectively:

$$\begin{aligned}
\Omega_\sigma &:= \operatorname{int} \bigcup \{\operatorname{clos} Q_i \,|\, \operatorname{meas}_d(Q_i \cap \Omega) > 0\},\\
\Omega_\sigma^\Gamma &:= \operatorname{int} \bigcup \{\operatorname{clos} Q_i \,|\, Q_i \in \Omega_\sigma \wedge Q_i \not\subset \Omega\},\\
\Omega_\sigma^\circ &:= \operatorname{int} \bigcup \{\operatorname{clos} Q_i \,|\, Q_i \in \Omega_\sigma \wedge Q_i \subset \Omega\},
\end{aligned} \tag{4.33}$$

with Q_i as in (4.29). Here, we write under a slight abuse of notation $Q_i \in \Omega_\sigma^\Gamma$ if $Q_i \subset \Omega_\sigma^\Gamma$. These domains obviously fulfil $\Omega_\sigma^\circ \subset \Omega \subset \Omega_\sigma$, $\Omega_\sigma = \operatorname{int}(\operatorname{clos}\Omega_\sigma^\circ \cup \operatorname{clos}\Omega_\sigma^\Gamma)$, and $\Omega_\sigma^\circ \cap \Omega_\sigma^\Gamma = \emptyset$. Two elements Q_i and Q_i' will be called neighbours if they share at least one node on the Cartesian grid. We will make the following somewhat technical assumption: for every $Q_i \in \Omega_\sigma^\Gamma$ there is a finite sequence of elements $(Q_i = Q_{i,1}, Q_{i,2}, \ldots, Q_{i,K}) \subset \Omega_\sigma^\Gamma$ with the following properties: the number K is bounded independent of σ, every pair of two subsequent elements are neighbours, and $Q_{i,K}$ has a neighbour in Ω_σ°. This condition means that one can always reach uncut elements from cut elements in a bounded number of steps. For sufficiently fine Cartesian grids this condition is often fulfilled with $K = 1$; if necessary it can be enforced by moving additional elements from Ω_σ° to Ω_σ^Γ.

4.3.2 Introduction of a Higher-order Stabilization Term

The basic idea of the ghost penalty method is to control the norm of cut elements by relating them to neighboring uncut elements. In the aforementioned articles [86, 87, 88], for example, this is done by controlling the norms of the gradient-jumps at element boundaries. However, as our PUFEM spaces are globally smooth, they do not contain such jumps. Burman and Fernández [91] and Cattaneo et al. [90] instead use

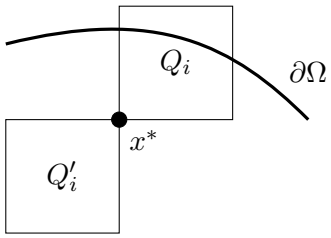

Figure 4.3: A cut element $Q_i \in \Omega_\sigma^\Gamma$ sharing a node x^* with an uncut element $Q_i' \in \Omega_\sigma^\circ$.

the Brezzi–Pitkäranta stabilisation [92]. We will use a higher-order variant of this idea and define the following bilinear form:

$$j(u_\sigma, v_\sigma) := \sigma^{2(P+1)} \sum_{Q_i \in \Omega_\sigma^\Gamma} \sum_{|\alpha|=P+1} \int_{Q_i} (\partial^\alpha u_\sigma)(\partial^\alpha v_\sigma) \,\mathrm{d}x, \tag{4.34}$$

such that $j(u_\sigma, u_\sigma) = \sigma^{2(P+1)} |u_\sigma|^2_{W^{P+1,2}(\Omega_\sigma^\Gamma)}$. We then obtain the following result:

Lemma 4.10. *Let $u_\sigma \in V_\sigma^P(\Omega_\sigma)$. One then has with constants c and C independent of σ and the position of $\partial\Omega$ relative to the Cartesian grid:*

$$c\|u_\sigma\|^2_{L^2(\Omega_\sigma)} \leq \|u_\sigma\|^2_{L^2(\Omega_\sigma^\circ)} + j(u_\sigma, u_\sigma) \leq C\|u_\sigma\|^2_{L^2(\Omega_\sigma)}. \tag{4.35}$$

Before moving on to the proof of this lemma, let us remark that the stabilisation term is necessary. Look for example at the configuration shown in figure 4.3. The partition of unity function corresponding to the node of Q_i opposite to x^* vanishes on Ω_σ°. Thus its $L^2(\Omega_\sigma)$-norm cannot be controlled by looking at Ω_σ° only, unless one adds a stabilisation term.

Proof. The second inequality directly follows from the inverse inequalities (4.32). For the first inequality, let us first consider the case $K = 1$, i. e., a cut element $Q_i \in \Omega_\sigma^\Gamma$ and an associated uncut element $Q_i' \in \Omega_\sigma^\circ$ which share a Cartesian grid point x^*, as for example illustrated in figure 4.3. This configuration can be mapped to one of $P_2^{2^d} = 4^d - 2^d$ reference cases with reference elements $\hat{Q}$ and $\hat{Q}'$ using the transformation $\hat{x} = \Phi^{-1}(x) := (x - x^*)/\sigma$, such that $\hat{x}^* = 0$. For an arbitrary function $v_\sigma \in V_\sigma^P(Q_i \cup Q_i')$ one obtains with $\hat{v} := (v_\sigma \circ \Phi) \in V_1^P(\hat{Q} \cup \hat{Q}')$:

$$\|v_\sigma\|^2_{L^2(Q_i \cup Q_i')} = \sigma^d \|\hat{v}\|^2_{L^2(\hat{Q} \cup \hat{Q}')}. \tag{4.36}$$

We claim that the following expression constitutes a norm on $V_1^P(\hat{Q} \cup \hat{Q}')$:

$$\|\hat{v}\|^2_* := \|\hat{v}\|^2_{L^2(\hat{Q}')} + |\hat{v}|^2_{W^{P+1,2}(\hat{Q})}. \tag{4.37}$$

It suffices to show that $\|\hat{v}\|_* = 0 \Longrightarrow \hat{v} = 0$. From $\|\hat{v}\|^2_{L^2(\hat{Q}')} = 0$ we obtain $\hat{v} \equiv 0$ on $\hat{Q}'$ and due to the global smoothness of $\hat{v}$ also $\partial^\alpha \hat{v}(\hat{x}^*) = 0$ for all multi-indices $\alpha \in \mathbb{N}_0^d$. From $|\hat{v}|^2_{W^{P+1,2}(\hat{Q})} = 0$ we obtain $\partial^\alpha \hat{v} \equiv 0, |\alpha| = P + 1$ on $\hat{Q}$. Together with

$\partial^\alpha \hat{v}(\hat{x}^*) = 0$ this implies $\hat{v} \equiv 0$ on $\hat{Q}$ as well. Thus $\|\cdot\|_*$ is indeed a norm. After employing the norm-equivalence of finite-dimensional spaces, we can transform back to $Q_i \cup Q_i'$ and obtain:

$$\sigma^d \|\hat{v}\|^2_{L^2(\hat{Q}\cup\hat{Q}')} \leq C\sigma^d \|\hat{v}\|^2_* = C\big(\|v_\sigma\|^2_{L^2(Q_i')} + \sigma^{2(P+1)} |v_\sigma|^2_{W^{P+1,2}(Q_i)}\big). \tag{4.38}$$

The case $K > 1$ with sequences of cells $(Q_i = Q_{i,1}, Q_{i,2}, \ldots, Q_{i,K} = Q_i')$ follows by induction. Now, summing over all elements and using the finite overlap condition $M = 2^d$, the claim follows. □

Note that the proof crucially depends on the global smoothness of the spaces V_σ^P. In particular, this stabilisation does not work with the conventional finite element spaces. As an example consider the case depicted in figure 4.3, and set $v_\sigma := 0$ on Q_i', and $v_\sigma := (x - x^*) \cdot y$ on Q_i, where $y \in \mathbb{R}^d$ is an arbitrary non-zero vector.

4.3.3 Stability and Convergence

As described before, we are aiming to mimic the $L^2(\Omega)$-orthogonal projector. To this end, we introduce the bilinear forms a and A:

$$a : L^2(\Omega_\sigma) \times L^2(\Omega_\sigma) \to \mathbb{R}, \qquad (u, v) \mapsto \int_\Omega uv \,\mathrm{d}x, \tag{4.39}$$

$$A : W^{P+1,2}(\Omega_\sigma) \times W^{P+1,2}(\Omega_\sigma) \to \mathbb{R}, \qquad (u_\sigma, v_\sigma) \mapsto a(u_\sigma, v_\sigma) + \varepsilon j(u_\sigma, v_\sigma), \tag{4.40}$$

where $\varepsilon > 0$ denotes a user-defined stabilisation parameter. We define the variational problem as: given any u for which the following makes sense, find $u_\sigma \in V_\sigma^P(\Omega_\sigma)$ such that:

$$A(u_\sigma, v_\sigma) = \int_\Omega u v_\sigma \,\mathrm{d}x \qquad \forall v_\sigma \in V_\sigma^P(\Omega_\sigma). \tag{4.41}$$

We then obtain the following two results.

Theorem 4.11 (Stability). *The bilinear form A as defined above fulfils with a constant $C(\varepsilon)$ independent of σ, u_σ, and the position of $\partial\Omega$ relative to the grid:*

$$A(u_\sigma, u_\sigma) \geq C(\varepsilon) \|u_\sigma\|^2_{L^2(\Omega_\sigma)} \quad \forall u_\sigma \in V_\sigma^P(\Omega_\sigma). \tag{4.42}$$

Proof. For any $u_\sigma \in V_\sigma^P(\Omega_\sigma)$ one has with the help of (4.35):

$$A(u_\sigma, u_\sigma) = \|u_\sigma\|^2_{L^2(\Omega)} + \varepsilon j(u_\sigma, u_\sigma) \geq \\ \|u_\sigma\|^2_{L^2(\Omega_\sigma^\circ)} + \varepsilon j(u_\sigma, u_\sigma) \geq C(\varepsilon) \|u_\sigma\|^2_{L^2(\Omega_\sigma)}. \tag{4.43}$$

□

Theorem 4.12 (Convergence). *Let $u \in W^{P+1,2}(\Omega_\sigma)$. The solution $u_\sigma \in V_\sigma^P(\Omega_\sigma)$ of the variational problem (4.41) then satisfies the following error bound:*

$$\|u_\sigma - u\|_{L^2(\Omega_\sigma)} \leq C(\varepsilon) \sigma^{P+1} \|u\|_{W^{P+1,2}(\Omega_\sigma)}, \tag{4.44}$$

where the constant $C(\varepsilon)$ is independent of σ, u, and how the boundary $\partial\Omega$ intersects the grid.

Proof. According to theorem 4.4, there exists $\mathcal{P}u \in V_\sigma^P(\Omega_\sigma)$ such that:

$$|\mathcal{P}u - u|_{W^{k,2}(\Omega_\sigma)} \leq C\sigma^{P+1-k}|u|_{W^{P+1,2}(\Omega_\sigma)} \qquad k \in \mathbb{N}_0,\ k \leq P+1. \tag{4.45}$$

We may write:

$$\|u_\sigma - u\|_{L^2(\Omega_\sigma)} \leq \|u_\sigma - \mathcal{P}u\|_{L^2(\Omega_\sigma)} + \|\mathcal{P}u - u\|_{L^2(\Omega_\sigma)}. \tag{4.46}$$

For the second term we can apply relation (4.45). For the first term we obtain with theorem 4.11 and the fact that u_σ solves (4.41):

$$\begin{aligned}\|\mathcal{P}u - u_\sigma\|^2_{L^2(\Omega_\sigma)} &\leq C(\varepsilon)A(\mathcal{P}u - u_\sigma, \mathcal{P}u - u_\sigma) = \\ &C(\varepsilon)\Big(\left(\mathcal{P}u - u, \mathcal{P}u - u_\sigma\right)_{L^2(\Omega)} + \varepsilon j(\mathcal{P}u, \mathcal{P}u - u_\sigma)\Big) \leq \\ C(\varepsilon)\Big(&\|\mathcal{P}u - u\|_{L^2(\Omega_\sigma)}\|\mathcal{P}u - u_\sigma\|_{L^2(\Omega_\sigma)} + \varepsilon j(\mathcal{P}u, \mathcal{P}u)^{1/2} j(\mathcal{P}u - u_\sigma, \mathcal{P}u - u_\sigma)^{1/2}\Big),\end{aligned} \tag{4.47}$$

where we used the Cauchy–Schwarz inequality in the last step. Noting that by the inverse estimates (4.32) we have:

$$j(\mathcal{P}u - u_\sigma, \mathcal{P}u - u_\sigma)^{1/2} \leq C\|\mathcal{P}u - u_\sigma\|_{L^2(\Omega_\sigma)} \tag{4.48}$$

and together with (4.45):

$$j(\mathcal{P}u, \mathcal{P}u)^{1/2} \leq C\sigma^{P+1}\|\mathcal{P}u\|_{W^{P+1,2}(\Omega_\sigma)} \leq C\sigma^{P+1}\|u\|_{W^{P+1,2}(\Omega_\sigma)}. \tag{4.49}$$

After dividing both sides by $\|\mathcal{P}u - u_\sigma\|_{L^2(\Omega_\sigma)}$ we thus obtain:

$$\|\mathcal{P}u - u_\sigma\|_{L^2(\Omega_\sigma)} \leq C(\varepsilon)\Big(\|\mathcal{P}u - u\|_{L^2(\Omega_\sigma)} + \varepsilon\sigma^{P+1}\|u\|_{W^{P+1,2}(\Omega_\sigma)}\Big). \tag{4.50}$$

Again applying (4.45) to the first term yields the claim. □

4.3.4 Influence of the Quadrature Error

In vortex methods we are only given a particle field, i. e., a quadrature rule for integrating smooth functions against the underlying vorticity we are aiming to approximate. Furthermore the bilinear form A can usually only be computed approximately, using numerical quadrature. In this subsection we are analysing the influence of these additional sources of error.

We will assume that the bilinear form j can be computed exactly. This is justified as it is sufficient to perform computations on the reference element $\hat{Q}$, which can be done up to arbitrary precision a priori. As for the bilinear form a, we will assume the availability of quadrature rules I_m satisfying error bounds of the following form:

$$\left|\int_\Omega f \,\mathrm{d}x - I_m(f)\right| \leq Ch^{m+1}|f|_{W^{m+1,1}(\Omega)} \quad f \in W^{m+1,1}(\Omega). \tag{4.51}$$

Error bounds of this form arise for example from composite quadrature rules of exactness degree m on quasi-uniform meshes of mesh width h, see theorem 2.3. Note that due to the global smoothness of the PUFEM spaces, these quadrature rules *do not need to be aligned* with the Cartesian grid. We will write $a_h(u_\sigma, v_\sigma) := I_m(u_\sigma v_\sigma)$ for the resulting approximate bilinear form. For $u_\sigma, v_\sigma \in V_\sigma^P(\Omega_\sigma)$ one then obtains with the help of Hölder's inequality and the inverse estimates (4.32):

$$\begin{aligned}|a(u_\sigma, v_\sigma) - a_h(u_\sigma, v_\sigma)| \leq Ch^{m+1}|u_\sigma v_\sigma|_{W^{m+1,1}(\Omega_\sigma)} \leq \\ Ch^{m+1} \sum_{|\alpha|=m+1} \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \|\partial^\beta u_\sigma \partial^{\alpha-\beta} v_\sigma\|_{L^1(\Omega_\sigma)} \leq \\ Ch^{m+1} \sum_{|\alpha|=m+1} \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} \|\partial^\beta u_\sigma\|_{L^2(\Omega_\sigma)} \|\partial^{\alpha-\beta} v_\sigma\|_{L^2(\Omega_\sigma)} \leq \\ Ch^{m+1}\sigma^{-(m+1)} \|u_\sigma\|_{L^2(\Omega_\sigma)} \|v_\sigma\|_{L^2(\Omega_\sigma)}. \quad (4.52)\end{aligned}$$

This will require us to couple h and σ through a relation like $h = \sigma^s$, for some $s > 1$. We then obtain coercivity of $A_h(u_\sigma, v_\sigma) := a_h(u_\sigma, v_\sigma) + j(u_\sigma, v_\sigma)$:

$$\begin{aligned}A_h(u_\sigma, u_\sigma) = A(u_\sigma, u_\sigma) - \big(a(u_\sigma, u_\sigma) - a_h(u_\sigma, u_\sigma)\big) \geq \\ \big(C(\varepsilon) - Ch^{m+1}\sigma^{-(m+1)}\big)\|u_\sigma\|^2_{L^2(\Omega_\sigma)} \geq C(\varepsilon)\|u_\sigma\|^2_{L^2(\Omega_\sigma)}, \quad (4.53)\end{aligned}$$

where the last constant $C(\varepsilon)$ is independent of σ, u, and the position of $\partial\Omega$ relative to the Cartesian grid, for $\sigma > 0$ small enough, $h = \sigma^s$, $s > 1$.

For the particle field u_h we will assume an error bound of the following form:

$$\|u_h - u\|_{W^{-(m+1),2}(\Omega)} \leq Ch^{m+1}\|u\|_{W^{m+1,2}(\Omega)}, \quad (4.54)$$

which is the typical form arising in vortex methods, see for example theorem 2.4. Again, the particle field does not in any way need to be aligned to the Cartesian grid. Collecting all of the previous results, we are ready to prove the main result of this chapter.

THEOREM 4.13. *Let $h = \sigma^s$, $s > 1$, and denote $k := \max\{P, m\}$. Let $u \in W^{k+1,2}(\Omega)$ and let the particle approximation $u_h \in W^{-(m+1),2}(\Omega)$ satisfy the error bound* (4.54)*. Then for $\sigma > 0$ small enough the solution $u_\sigma \in V_\sigma^P(\Omega_\sigma)$ of the perturbed variational problem:*

$$A_h(u_\sigma, v_\sigma) = \langle u_h, v_\sigma \rangle \quad \forall v_\sigma \in V_\sigma^P(\Omega_\sigma) \quad (4.55)$$

satisfies the following error bound:

$$\|u - u_\sigma\|_{L^2(\Omega)} \leq C(\varepsilon)\big(\sigma^{P+1} + h^{m+1}\sigma^{-(m+1)}\big)\|u\|_{W^{k+1,2}(\Omega)}, \quad (4.56)$$

where the constant $C(\varepsilon)$ is independent of σ, u, and the position of $\partial\Omega$ relative to the Cartesian grid.

Proof. Let $v_\sigma \in V_\sigma^P(\Omega_\sigma)$ denote the solution of the unperturbed variational problem (4.41), with u extended to $\mathcal{E}u$ by the Stein extension operator. With the help of the coercivity of A_h one then obtains:

$$\|u_\sigma - v_\sigma\|^2_{L^2(\Omega_\sigma)} \leq C(\varepsilon) A_h(u_\sigma - v_\sigma, u_\sigma - v_\sigma) = \\ C(\varepsilon)\Big(\langle u_h - u, u_\sigma - v_\sigma\rangle + A(v_\sigma, u_\sigma - v_\sigma) - A_h(v_\sigma, u_\sigma - v_\sigma)\Big). \quad (4.57)$$

Application of the error bounds (4.54) and (4.52) as well as the inverse estimates (4.32) yields:

$$\|u_\sigma - v_\sigma\|^2_{L^2(\Omega_\sigma)} \leq C(\varepsilon)\big(h^{m+1}\sigma^{-(m+1)}\|u\|_{W^{m+1,2}(\Omega)}\|u_\sigma - v_\sigma\|_{L^2(\Omega_\sigma)} + \\ h^{m+1}\sigma^{-(m+1)}\|v_\sigma\|_{L^2(\Omega_\sigma)}\|u_\sigma - v_\sigma\|_{L^2(\Omega_\sigma)}\big). \quad (4.58)$$

and thus:

$$\|u_\sigma - v_\sigma\|_{L^2(\Omega_\sigma)} \leq C(\varepsilon) h^{m+1}\sigma^{-(m+1)}\big(\|u\|_{W^{m+1,2}(\Omega)} + \|v_\sigma\|_{L^2(\Omega_\sigma)}\big). \quad (4.59)$$

Nothing that $\|v_\sigma\|_{L^2(\Omega_\sigma)} \leq C(\varepsilon)\|u\|_{W^{P+1,2}(\Omega)}$ we obtain:

$$\|u_\sigma - v_\sigma\|_{L^2(\Omega_\sigma)} \leq C(\varepsilon) h^{m+1}\sigma^{-(m+1)}\|u\|_{W^{k+1,2}(\Omega)}. \quad (4.60)$$

Now, by the triangle inequality:

$$\|u_\sigma - u\|_{L^2(\Omega)} \leq \|u_\sigma - v_\sigma\|_{L^2(\Omega)} + \|v_\sigma - u\|_{L^2(\Omega)} \quad (4.61)$$

the claim follows by applying theorem 4.12 and the boundedness of $\mathcal{E}$ to the second term. □

The part $\sigma^{P+1}\|u\|_{W^{k+1,2}(\Omega)}$ is called the smoothing error; σ roughly corresponds to the blob-width in conventional vortex particle methods. The second part is called the quadrature error; choosing $s = 1 + \frac{P+1}{m+1}$ balances both terms. In the next section we will illustrate that the choice $P = m$ does not only 'feel natural', but also yields optimal results in a certain sense. In this case we obtain with $s = 2$ an overall convergence rate of $\mathcal{O}(\sigma^{P+1}) = \mathcal{O}(h^{\frac{1}{2}(P+1)})$.

4.3.5 Optimality of the Smoothed Solution

In this subsection we will assume that we can apply the bilinear form A exactly, i. e., $A_h = A$. Furthermore we assume $P = m$. We will show that the smoothed solution u_σ then satisfies the same asymptotic error bound as u_h. We will need the following corollary of theorem 4.12.

Corollary 4.14. *The solution operator $\mathcal{S}$ to the problem* (4.41) *is bounded:*

$$\|\mathcal{S}u\|_{W^{P+1,2}(\Omega_\sigma)} \leq C(\varepsilon)\|u\|_{W^{P+1,2}(\Omega_\sigma)} \qquad \forall u \in W^{P+1,2}(\Omega_\sigma). \quad (4.62)$$

Proof. One has $\|\mathcal{S}u\|_{W^{P+1,2}(\Omega_\sigma)} \leq \|\mathcal{S}u - u\|_{W^{P+1,2}(\Omega_\sigma)} + \|u\|_{W^{P+1,2}(\Omega_\sigma)}$. Using $\mathcal{P}$ as defined above, we furthermore obtain: $\|\mathcal{S}u - u\|_{W^{P+1,2}(\Omega_\sigma)} \leq \|\mathcal{S}u - \mathcal{P}u\|_{W^{P+1,2}(\Omega_\sigma)} + \|\mathcal{P}u - u\|_{W^{P+1,2}(\Omega_\sigma)}$. The second term can be bounded by $C\|u\|_{W^{P+1,2}(\Omega_\sigma)}$ due to the boundedness of $\mathcal{P}$. The first term can be bounded by first applying the inverse estimates (4.32) followed by estimate (4.50). □

Theorem 4.15 (Optimality). *Let the conditions of theorem 4.13 be fulfilled. Furthermore assume that $A_h = A$, $P = m$, and $s = 2$. Then the smoothed solution u_σ fulfills:*

$$\|u_\sigma - u\|_{W^{-(P+1),2}(\Omega)} \leq C(\varepsilon) h^{P+1} \|u\|_{W^{P+1,2}(\Omega)}. \tag{4.63}$$

Proof. Let $\varphi \in W^{P+1,2}(\Omega)$ be arbitrary but fixed. One has with $\mathcal{P}$ and $\mathcal{E}$ as defined above:

$$\int_\Omega (u_\sigma - u)\varphi \,\mathrm{d}x = \int_\Omega (u_\sigma - u)(\varphi - \mathcal{P}\mathcal{E}\varphi)\,\mathrm{d}x + \int_\Omega (u_\sigma - u)\mathcal{P}\mathcal{E}\varphi \,\mathrm{d}x. \tag{4.64}$$

For the first term we obtain with the Cauchy–Schwarz inequality, theorem 4.4, and theorem 4.13:

$$\begin{aligned}\int_\Omega (u_\sigma - u)(\varphi - \mathcal{P}\mathcal{E}\varphi)\,\mathrm{d}x &\leq \|u - u_\sigma\|_{L^2(\Omega)} \|\varphi - \mathcal{P}\mathcal{E}\varphi\|_{L^2(\Omega)} \\ &\leq C(\varepsilon)\sigma^{2(P+1)} \|u\|_{W^{P+1,2}(\Omega)} \|\varphi\|_{W^{P+1,2}(\Omega)}.\end{aligned} \tag{4.65}$$

For the second term one has:

$$\begin{aligned}\int_\Omega (u_\sigma - u)\mathcal{P}\mathcal{E}\varphi \,\mathrm{d}x = A(u_\sigma, \mathcal{P}\mathcal{E}\varphi) - \int_\Omega u\mathcal{P}\mathcal{E}\varphi \,\mathrm{d}x - \varepsilon j(u_\sigma, \mathcal{P}\mathcal{E}\varphi) = \\ \langle u_h - u, \mathcal{P}\mathcal{E}\varphi\rangle - \varepsilon j(u_\sigma, \mathcal{P}\mathcal{E}\varphi) \leq \\ C\Big(\|u_h - u\|_{W^{-(P+1),2}(\Omega)} \|\varphi\|_{W^{P+1,2}(\Omega)} + \varepsilon\sigma^{2(P+1)} \|u_\sigma\|_{W^{P+1,2}(\Omega_\sigma)} \|\varphi\|_{W^{P+1,2}(\Omega)}\Big).\end{aligned} \tag{4.66}$$

It remains to show that $\|u_\sigma\|_{W^{P+1,2}(\Omega_\sigma)} \leq C(\varepsilon)\|u\|_{W^{P+1,2}(\Omega)}$. To see this, note that we have:

$$\begin{aligned}\|u_\sigma\|_{W^{P+1,2}(\Omega_\sigma)} \leq \|\mathcal{S}\mathcal{E}u\|_{W^{P+1,2}(\Omega_\sigma)} + \|u_\sigma - \mathcal{S}\mathcal{E}u\|_{W^{P+1,2}(\Omega_\sigma)} \leq \\ C(\varepsilon)\|u\|_{W^{P+1,2}(\Omega)} + \|u_\sigma - \mathcal{S}\mathcal{E}u\|_{W^{P+1,2}(\Omega_\sigma)}.\end{aligned} \tag{4.67}$$

Applying inequality (4.60) to the second term, collecting all the terms, and noting that $\sigma = \sqrt{h}$ yields the result. □

4.3.6 Conservation Properties

In the introduction we mentioned the conservation properties of vortex methods as one of their highlights. In this section we make some brief remarks on some of these

properties under the assumption that $A_h = A$. For brevity, we will focus on the two-dimensional case, but remark that all of the results we present here analogously hold in three-dimensions.

The conserved quantities circulation, linear momentum, and angular momentum are given by $I_0 = \int_\Omega 1 \cdot \omega \,\mathrm{d}x$, $\mathbf{I}_1 = \int_\Omega (x_2, -x_1)^\top \omega \,\mathrm{d}x$, and $I_2 = \int_\Omega |x|^2 \omega \,\mathrm{d}x$, respectively [17, Section 1.7]. Noting that the stabilisation term j vanishes if one of its arguments is a polynomial of total degree less than P, one obtains for the solution ω_σ of (4.41) with right-hand side ω_h: $(\omega_\sigma, x^\alpha)_{L^2(\Omega)} = \omega_h(x^\alpha)$ for all $|\alpha| \leq P$. For $P = 1$ we consequently conserve I_0 and $\mathbf{I}_1$, for $P = 2$ one additionally conserves angular momentum I_2. This is important, because in vortex methods body forces are often computed using the relation $\mathbf{F} = -\rho \frac{\mathrm{d}\mathbf{I}_1}{\mathrm{d}t}$, where ρ denotes the fluid's density.

4.4 Numerical Experiments

We have now established the necessary results to return to our original motivation. Given a particle approximation $\omega_h \in W^{-(m+1),2}(\Omega)$ of the vorticity ω that satisfies an error-bound of the form $\|\omega_h - \omega\|_{W^{-(m+1),2}(\Omega)} \leq Ch^{m+1}\|\omega\|_{W^{m+1,2}(\Omega)}$, we want to obtain a smooth approximation ω_σ, such that we can compute the corresponding induced velocity field using the Biot–Savart law $\mathbf{u}_\sigma = \mathbf{K} \star \omega_\sigma$. One can then use this approximate velocity field to advance ω_h in time by convecting the particles according to $\frac{\mathrm{d}x_i}{\mathrm{d}t}(t) = \mathbf{u}_\sigma(t, x_i(t))$.

In section 4.2 we introduced the spaces $V_\sigma^P(\Omega)$ that can be used as test-spaces for the particle field ω_h. In section 4.3 we modelled the regularisation problem as a perturbation to a stabilised L^2-projection onto the spaces $V_\sigma^P(\Omega)$. The analysis indicated that one should choose $P = m$ and $\sigma = \sqrt{h}$, resulting in an a-priori error estimate of $\|\omega_\sigma - \omega\|_{L^2(\Omega)} \leq C\sigma^{P+1}\|\omega\|_{W^{P+1,2}(\Omega)}$. The Calderón–Zygmund inequality (4.4) then tells us that one may expect $\|\mathbf{u}_\sigma - \mathbf{u}\|_{W^{1,2}(\Omega)} \leq C\sigma^{P+1}\|\omega\|_{W^{P+1,2}(\Omega)}$ for the resulting velocity field $\mathbf{u}_\sigma := \mathbf{K} \star \omega_\sigma$. This analogously holds in the three-dimensional case.

In this section we perform several numerical experiments. We will first describe the experimental setup. We then perform experiments on a scalar particle field and confirm the results of our analysis. In particular, the experiments will show that the common practice of choosing σ proportional to h instead of $\sqrt{h}$ does not lead to convergent schemes. We will then illustrate the practicality of our scheme, by approximating a vector-valued vorticity field, computing its induced velocity field, and measuring the error. We finish this section with experiments on the condition number of the resulting systems and its dependence of the stabilisation parameter ε.

4.4.1 Setup

We define our computational test domain as $\Omega = (-\frac{1}{2}, \frac{1}{2})^3$. While this is one of the simplest cases for mesh based methods, due to its sharp corners and edges it is one of the hardest for conventional vortex blob methods. In order to obtain quadrature rules which are not aligned to the Cartesian grid, the mesh generator Gmsh [58] was used to

obtain a tetrahedral mesh of the domain, consisting of 24 tetrahedra with maximum edge-length $h = 1$. The quadrature rules are obtained by applying the mid-point rule to this mesh and its subsequent uniform refinements from level $l = 0$ down to level $l = 8$, corresponding to $h = 2^{-8} \approx 0.004$ and $N = 402\,653\,184$ quadrature nodes.

Preliminary experiments showed good results for a stabilisation parameter of $\varepsilon = 0.001$, which we will use for all of our computations. We will use degree $P = 1$ for the PUFEM spaces, set $\sigma := Ch^{1/s}$, and experiment on various choices of C and $s = 1, 2$. For the integration of the bilinear form A_h we use the following approach: if in a pair of basis functions one of them has cut support, we use the same quadrature rule as for the particle field. Otherwise precomputed values from the reference element $\hat{Q}$ are used. The resulting systems of equations are solved using the conjugate gradient method, where we apply a simple diagonal scaling as preconditioner. The iteration was stopped when a relative residual of 10^{-12} was reached. This was usually the case after less than 100 iterations, with some exceptions for coarse refinement levels l and the case $C = 0.5, s = 1$.

4.4.2 Scalar Particle Field

The common practice to choose the smoothing length σ proportional to h may in special cases be justified with the analysis of Cottet and Koumoutsakos [48, Section 2.6]. They assume that the quadrature rules used are of infinite order, essentially corresponding to the case $m = \infty$. Such rules, however, typically only exist in very special cases, such as a cube with periodic or zero boundary conditions. To show that this approach does not work in a more general setting, we aim to approximate the following function:

$$u(\mathbf{x}) = \cos(4\pi x_1) \qquad \mathbf{x} \in (-\tfrac{1}{2}, \tfrac{1}{2})^3. \tag{4.68}$$

This function *does not vanish at the boundary.* The application of conventional blob-methods would thus blur the boundaries and lead to only slowly converging schemes. We define the particle field as $u_h := \sum_{i=1}^{N} w_i u(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)$, with δ denoting the Dirac Delta, and $\mathbf{x}_i$ and w_i being the positions and weights of the mid-point quadrature rule applied to the tetrahedra of the mesh at various refinement levels as defined above.

Figure 4.4 shows the error $\|u - u_\sigma\|_{L^2(\Omega)}$ for $\sigma = Ch$ for various choices of C at different refinement levels. Choosing $C = 0.5$ results in approximations with large errors, which do not decrease significantly under mesh refinement. The case $l = 8$ was not computed due to the large memory requirements. The other curves exhibit similar behaviour: in the beginning and intermediate stages the error decreases, however, only at an approximately linear, not quadratic rate. This rate further decreases and approaches zero under mesh refinement, confirming the predicted bound of the quadrature error $\mathcal{O}(h^{m+1}\sigma^{-(m+1)}) = \mathcal{O}(1)$. Choosing larger values C somewhat delays but does not prevent this effect, at the cost of larger errors on coarse refinement levels.

Figure 4.5 shows the corresponding error for the case $\sigma = C\sqrt{h}$. All choices of C lead to convergent schemes which approach the predicted convergence rate of $\mathcal{O}(h)$. In our experiments, smaller choices of C lead to smaller errors; however choosing C

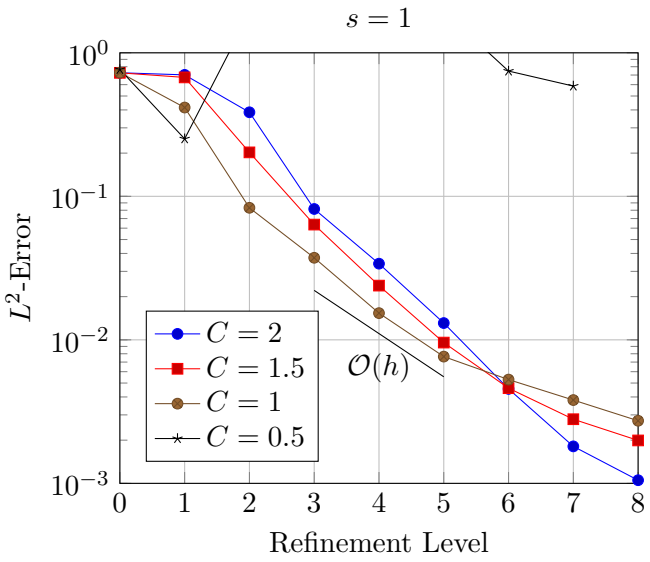

Figure 4.4: L^2-Error of the smoothed approximation u_σ in the case $\sigma = Ch$.

too small causes larger errors in the coarser cases. In our test case a choice somewhere between $C = 0.25$ and $C = 0.5$ seems to be optimal.

4.4.3 Vector-valued Particle Field and Velocity Evaluation

In this section we show that our scheme can drastically reduce the cost of the computationally most expensive part of vortex methods, the velocity evaluation. To this end, we prescribe:

$$\mathbf{u}(\mathbf{x}) := \begin{pmatrix} x_2 \\ -x_1 \\ 0 \end{pmatrix} \exp\left(-\frac{1}{1-4|\mathbf{x}|^2}\right) \qquad \mathbf{x} \in (-\tfrac{1}{2}, \tfrac{1}{2})^3. \tag{4.69}$$

This velocity field is smooth and fulfils $\nabla \cdot \mathbf{u} \equiv 0$. It was chosen such that it vanishes at the boundaries, so that it can be retrieved from the vorticity field $\boldsymbol{\omega} := \nabla \times \mathbf{u}$ through the Biot–Savart law without any boundary integral terms:

$$\mathbf{u}(\mathbf{x}) = -\frac{1}{4\pi} \int_\Omega \frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^3} \times \boldsymbol{\omega}(\mathbf{y})\, \mathrm{d}\mathbf{y}. \tag{4.70}$$

Analogous to the previous section, we define the particle approximation:

$$\boldsymbol{\omega}_h := \sum_{i=1}^{N} w_i \boldsymbol{\omega}(\mathbf{x}_i) \delta(\mathbf{x} - \mathbf{x}_i). \tag{4.71}$$

Experiments in the previous section suggested a choice of $\sigma = C\sqrt{h}$, with C between 0.25 and 0.5. We consequently choose $C = 0.375$ and obtain after applying the method

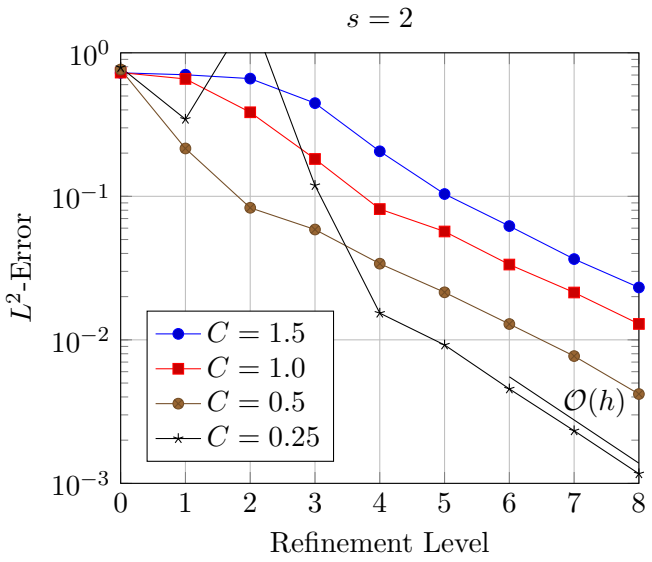

Figure 4.5: L^2-Error of the smoothed approximation u_σ in the case $\sigma = C\sqrt{h}$.

to each component a smoothed approximation $\boldsymbol{\omega}_\sigma$ with an anticipated convergence rate of $\mathcal{O}(\sigma^2)$ in the L^2-norm. In order to evaluate the Biot–Savart law for this vorticity field, we chose the coarsest level l such that the corresponding mesh width 2^{-l} is smaller than σ. We then compute the orthogonal projection of $\boldsymbol{\omega}_\sigma$ onto the *standard* finite element space of piecewise linear functions on that level. The Biot–Savart integral can then be computed by summing over the tetrahedra, for which the analytic formulae from chapter 5 can be used. We couple these formulae with a fast multipole method [43, 44] for the far-field evaluation. The resulting velocity field is approximated by taking the nodal interpolation onto the standard finite element space of piecewise *quadratics* to obtain an approximate velocity field $\mathbf{u}_\sigma$.

This scheme was chosen for its simplicity and speed. Most conventional schemes apply the fast multipole method directly to the particle field, leading to a complexity of $\mathcal{O}(N) = \mathcal{O}(h^{-d})$, with a large hidden constant. Note that in our case the method is applied to the coarser smoothed approximation, leading to a complexity of only $\mathcal{O}(h^{-\frac{d}{2}})$. Due to the interpolation onto the standard space of piecewise quadratics, however, the resulting approximate velocity $\mathbf{u}_\sigma$ does no longer fulfil $\nabla \cdot \mathbf{u}_\sigma = 0$. We will further comment on this issue on the concluding remarks of this chapter.

Figure 4.6 shows the L^2-errors in the approximate smoothed vorticity field $\boldsymbol{\omega}_\sigma$ and the velocity field $\mathbf{u}_\sigma$. The smoothed vorticity field converges at a rate of $\mathcal{O}(\sigma^2) = \mathcal{O}(h)$ as expected. With the Calderón–Zygmund inequality (4.4) we obtain that the same error bound holds for the velocity in the $W^{1,2}$-norm. As the results indicate, in the L^2-norm the error seems to reduce by one power in σ faster, resulting in a rate of $\mathcal{O}(h^{1.5})$.

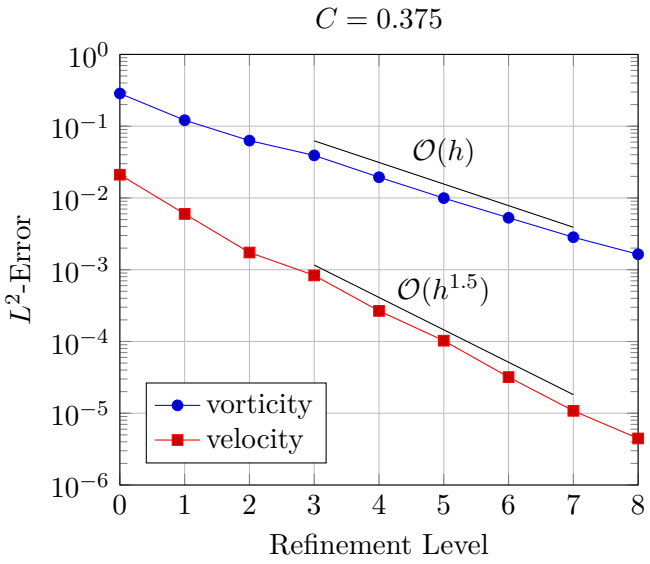

Figure 4.6: L^2-Error of the smoothed vorticity approximation $\boldsymbol{\omega}_\sigma$ and the resulting finite-element approximation of the corresponding velocity $\mathbf{u}_\sigma$.

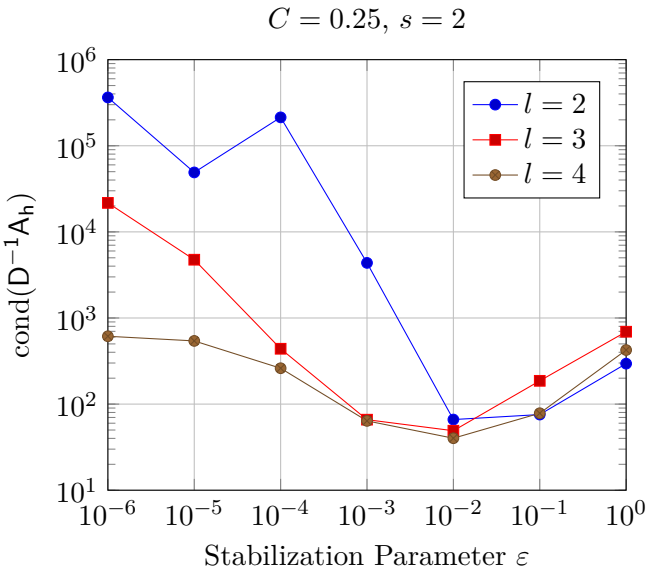

Figure 4.7: Condition number of the diagonally scaled system matrix $\mathsf{D}^{-1}\mathsf{A}_\mathsf{h}$.

4.4.4 System Condition Number

In this section we investigate the effect of the stabilisation parameter ε on the condition number of the system matrix. In subsection 4.4.2 we observed instabilities on the coarse levels l in the case $C = 0.25, s = 2$. We therefore chose this particular configuration for our experiments. We used the following set of functions as our basis for the spaces $V_\sigma^P(\Omega_\sigma)$:

$$\mathcal{B}_\sigma^P := \left\{ \varphi_i\left(\frac{x - x_i}{\sigma}\right)^\alpha \,\middle|\, Q_i \in \Omega_\sigma, |\alpha| \leq P \right\}. \tag{4.72}$$

We may assign a numbering $\mathcal{I} = \{1, \ldots, n\}$ to this set, and subsequently refer to its members as $\mathcal{B}_\sigma^P \ni \psi_k$, $k \in \mathcal{I}$. We can then define the system matrix $\mathsf{A_h} \in \mathbb{R}^{n \times n}$ via the relation:

$$\mathsf{e_k}^\top \mathsf{A_h} \mathsf{e_l} = A_h(\psi_k, \psi_l) = a_h(\psi_k, \psi_l) + \varepsilon j(\psi_k, \psi_l), \qquad \forall k, l \in \mathcal{I}, \tag{4.73}$$

where $\mathsf{e_k} \in \mathbb{R}^n$ refers to the k-th Cartesian basis vector, and the approximate bilinear form a_h is defined as described in the numerical setup (subsection 4.4.1). We are then interested in the condition number of the diagonally scaled matrix $\mathsf{D}^{-1}\mathsf{A_h}$, where $\mathsf{D} := \operatorname{diag} \mathsf{A_h}$.

Figure 4.7 shows the condition number of $\mathsf{D}^{-1}\mathsf{A_h}$ for various refinements levels l as a function of ε. In the case $l = 2$ the quadrature error is so large that the resulting matrix $\mathsf{A_h}$ ceased being positive definite for $\varepsilon = 10^{-3}$, and is even singular for $\varepsilon = 0$. This explains the large error observed in subsection 4.4.2 for this case. But even then a sufficiently large choice of ε results in a well conditioned system. For the finer refinement levels a choice of ε between 10^{-3} and 10^{-1} seems to be optimal and reduces the matrix' condition number below 100. The effect becomes slightly less pronounced with increasing l. We can thus conclude that for such a choice of ε the stabilisation removes the ill-conditioning of the system, especially in the presence of moderate quadrature errors.

4.5 Conclusions and Outlook

We have presented a new method to tackle the particle regularisation problem, based on a stabilised fictitious domain formulation with smooth shape-functions. Our approach enjoys all the benefits of the conventional blob-methods: the resulting smoothed approximations are C^∞ functions and conserve all moments up to order P. On top of that, our approach can accurately handle general geometries. The evaluation of the smoothed approximations is cheap and straightforward and does not require a summation over all particles as in the case of blob functions. Also note that the so-called 'point-location problem' is trivial for the spaces defined in this chapter: given any point $\mathbf{x} \in \Omega$ finding the cell $Q_i \ni \mathbf{x}$ that contains this point is trivial and amounts to a division of the point's Cartesian coordinates by σ. This problem is significantly harder for conventional meshes.

The fact that we can only achieve a convergence rate of $\mathcal{O}(h^{\frac{m+1}{2}})$ as opposed to $\mathcal{O}(h^{m+1})$ might seem disappointing, but is intrinsic to the smoothing problem at hand. This can be illustrated in a simple one-dimensional example: given an interval of length h on the real line, the m-node Gaussian quadrature rule will have an error bound of $\mathcal{O}(h^{2m})$. With m function values, however, we can only construct an interpolation polynomial of degree $m-1$, having the halved error bound $\mathcal{O}(h^m)$. Theorem 4.15 shows that the smoothed approximation is essentially just as accurate as the particle field. Also note that in a computer implementation handling particle fields is significantly simpler than handling conventional meshes: because of the lack of connectivity between particles, one only needs to store the particle locations and strengths in an array. This way even low-cost desktop computers can manage hundreds of millions of particles in memory. Subsequently h can be chosen much smaller than in mesh based approaches.

Because the smooth approximations $\boldsymbol{\omega}_\sigma$ fulfil the same asymptotic error bound as the given particle field, they can be used for a variety of purposes. As a simple example, $\boldsymbol{\omega}_\sigma$ could be used to reinitialise overly distorted particle fields. But even more important, the smooth approximations allow for a coupling of particle methods with mesh based methods, similar to Vortex-in-Cell schemes (VIC). This opens the door to a variety of hybridisation possibilities: one could for example solve the diffusion problem using $\boldsymbol{\omega}_\sigma$ and a finite element based method, while relying on the particle field $\boldsymbol{\omega}_h$ for the treatment of the convection.

A further consequence of the relation $h \sim \sigma^2$ is that the smoothed vorticity field $\boldsymbol{\omega}_\sigma$ has much greater length-scales than the particle spacing. As a consequence the velocity evaluation—usually the most expensive part of vortex methods—can be drastically sped up. In our numerical experiments we gave an example of a mesh based scheme for this, which was chosen for both its simplicity and speed. We believe that this scheme can be further improved. For example, we know that $\mathbf{K} \star \boldsymbol{\omega} = \nabla \times \mathbf{A}$, with $\mathbf{A}$ corresponding to the volume integral in (2.20). The *curl spaces* $\nabla \times (V_\sigma^P(\Omega)^3)$ would thus be natural candidates for ansatz spaces for the velocity approximation. These spaces would be divergence free in the strong, point-wise sense and thus maintain this desirable property of the Biot–Savart law.

It is not clear whether the exact variational formulation (4.41) is actually unstable without stabilisation. A result by Reusken [93, Theorem 5] indicates that a rescaling might be sufficient to achieve a stable formulation. On the other hand, this result assumes that the bilinear form a can be computed exactly. The experiments of subsection 4.4.4 indicate that stabilisation is especially beneficial in the presence of quadrature errors.

Our current approach uses a uniform grid size σ and a fixed polynomial degree P. The partition of unity approach, however, is general enough to be extended to adaptive grids and varying polynomial degrees. For the future, experiments with σ-, P-, or σP-adaptive schemes are another interesting field for further research.

Chapter 5

Evaluation of the Biot–Savart Law

In this chapter we describe how the Biot–Savart law from subsection 2.2.1 can be approximated using a quasi-uniform triangulation $\mathcal{G}_\sigma$ of the computational domain Ω. When used together with a particle regularisation scheme such as that from chapter 4 this enables us to approximate the velocity induced by a given particle approximation of the vorticity.

5.1 Introduction

In the previous chapter we described a particle regularisation scheme for general bounded domains $\Omega \subset \mathbb{R}^d$. Let us for example consider the case $d = 3$. Given particle field approximations $\boldsymbol{\omega}_h$ we can thus obtain corresponding smooth approximations $\boldsymbol{\omega}_\sigma$ that satisfy optimal error bounds of the form $\|\boldsymbol{\omega}_\sigma - \boldsymbol{\omega}\|_{L^2(\Omega)} \leq C\sigma^{P+1}\|\boldsymbol{\omega}\|_{W^{P+1,2}(\Omega)}$, with $h \sim \sigma^2$. These smooth approximations no longer arise from mollifications of a particle field with a radially symmetric blob-function ζ_σ, as it is the case for the conventional vortex blob methods. There thus no longer is a smoothed kernel $\mathbf{K}_\sigma$ such that the Biot–Savart law can be evaluated as $\mathbf{K}_\sigma \star \boldsymbol{\omega}_h$.

In the three-dimensional case we now are faced with the task of evaluating integrals of the form:

$$\frac{1}{4\pi}\int_\Omega \frac{\mathbf{x}-\mathbf{y}}{|\mathbf{x}-\mathbf{y}|^3} \times \boldsymbol{\omega}_\sigma(\mathbf{y})\,\mathrm{d}V(\mathbf{y}), \tag{5.1}$$

and analogously (2.13) for the two-dimensional case. For $\mathbf{x} \in \Omega$ the integrand of these integrals becomes singular, and the standard quadrature approaches from section 2.4 for smooth functions fail to give good results. The evaluation of such singular integrals thus requires special care.

Similarly to the problem of integrating smooth functions, evaluating singular integrals over arbitrary domains is a hard problem. As described in subsection 2.4.1, a common strategy to tackle this problem is to consider quasi-uniform, shape-regular triangulations $\mathcal{G}_\sigma$ of mesh-width σ of the domain Ω. This way the task of evaluating (5.1) can be reduced to integrating over tetrahedra, or triangles in the two-dimensional case, respectively. Similar to quadrature rules for smooth functions, one then aims to find formula that integrate all polynomial $\boldsymbol{\omega}_\sigma$ of a specified maximum degree P over cells T exactly. In this chapter we will cover the case $P = 1$. Extensions to higher order polynomials are possible in principle, however significantly more involved.

The two-dimensional case has already received considerable amount of study. Formulae for triangular cells and $P = 1$ have for example been published by Russo and Strain [80] in the context of their triangulated vortex method. The three-dimensional case seems to have received much less attention. The only formulae for the case $P = 1$ known to us are those by Suh [94]. They require the evaluation of six logarithms and twelve arcsines. While correct, they have the disadvantage that they depend on the coordinate system used. Additionally, they discontinuously depend on the position of the field-point relative to the source tetrahedron. In particular, they contain a factor α that jumps from 4π to 0 as the field-point $\mathbf{x}$ moves from outside the tetrahedron to its inside. On the boundary its value depends on whether $\mathbf{x}$ coincides with one of the tetrahedron's nodes or lies on one of its edges or faces. Thus, finding the correct α is difficult if floating-point arithmetic is used and $\mathbf{x}$ lies very close to or on the tetrahedron's boundary. This is quite unsatisfactory, as the resulting function $\mathbf{u}$ is in fact continuous. Problems with the formulae of Suh have also been reported by Li and Vezza [95].

In this chapter we illustrate how the Biot–Savart integral can be expressed by single layer potentials over the tetrahedron's faces. These integrals can then evaluated using the formulas of van Oosterom [96]. The resulting expressions are simpler, numerically stable, and cut the number of inverse trigonometric function computations from twelve to four.

5.2 Reduction to Single Layer Potentials

Let $T \subset \mathbb{R}^3$ be a non-degenerate tetrahedron and let $\mathbf{x} \in \mathbb{R}^3$ be an arbitrary but fixed point in space. Denote $\mathbf{r} = \mathbf{r}(\mathbf{y}) = \mathbf{y} - \mathbf{x}$, $r = |\mathbf{r}|$, and let F_i and $\mathbf{n}_i$, $i \in \{1, 2, 3, 4\}$, denote the faces of T and their respective exterior unit-normals. Furthermore, let $\boldsymbol{\omega}$ be a vector-valued function, where each component is a polynomial of total degree $P = 1$ or less. The Biot–Savart law $\mathbf{u} = \mathbf{K} \star \boldsymbol{\omega}$ with the kernel $\mathbf{K}$ from equation (2.14) can then equivalently be written as:

$$\mathbf{u}(\mathbf{x}) = -\frac{1}{4\pi} \int_T \nabla\left(\frac{1}{r}\right) \times \boldsymbol{\omega} \, \mathrm{d}V(\mathbf{y}), \tag{5.2}$$

where here and throughout this chapter the derivative is taken with respect to the $\mathbf{y}$ coordinate. Applying the vector calculus identity:

$$\nabla\left(\frac{1}{r}\right) \times \boldsymbol{\omega} = \nabla \times \left(\frac{\boldsymbol{\omega}}{r}\right) - \frac{\nabla \times \boldsymbol{\omega}}{r}, \tag{5.3}$$

one obtains with help of the divergence theorem:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{4\pi}\left(\int_T \frac{\nabla \times \boldsymbol{\omega}}{r} \, \mathrm{d}V(\mathbf{y}) + \int_{\partial T} \frac{\boldsymbol{\omega} \times \mathbf{n}}{r} \, \mathrm{d}S(\mathbf{y})\right), \tag{5.4}$$

where $\mathbf{n} = \mathbf{n}(\mathbf{y})$ denotes the unit exterior normal on T. For the remaining volume integral note that $\nabla \times \boldsymbol{\omega}$ is a constant and may be taken out of the integral. Noting

that:

$$\frac{2}{r} = \nabla \cdot \left(\frac{\mathbf{r}}{r}\right) \tag{5.5}$$

we then obtain for linearly varying $\boldsymbol{\omega}$ by again employing the divergence theorem:

$$\int_T \frac{\nabla \times \boldsymbol{\omega}}{r} \,\mathrm{d}V(\mathbf{y}) = \int_{\partial T} \frac{\frac{1}{2}(\nabla \times \boldsymbol{\omega})(\mathbf{r} \cdot \mathbf{n})}{r} \,\mathrm{d}S(\mathbf{y}). \tag{5.6}$$

And thus:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{4} \frac{1}{4\pi} \int_{F_i} \frac{\frac{1}{2}(\nabla \times \boldsymbol{\omega})(\mathbf{r} \cdot \mathbf{n}_i) + \boldsymbol{\omega} \times \mathbf{n}_i}{r} \,\mathrm{d}S(\mathbf{y}). \tag{5.7}$$

Note that for each face the enumerator now is a linearly varying function of $\mathbf{y}$. Denoting these functions with $\boldsymbol{\sigma}$ we have thus reduced the task to the evaluation of four single layer potentials over planar triangles. This will be the topic of the next section.

5.3 Evaluation of the Single Layer Potentials

Under the term *single layer potential* over a surface S we understand the following function:

$$(\mathcal{S}\boldsymbol{\sigma})(\mathbf{x}) := \frac{1}{4\pi} \int_S \frac{\boldsymbol{\sigma}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \,\mathrm{d}S(\mathbf{y}), \tag{5.8}$$

where $\boldsymbol{\sigma}$ is a given surface density. This function is one of the key building blocks of most boundary element methods; the boundary integrals in the Helmholtz decomposition in proposition 2.1 are also of this type. For this reason the evaluation of this integral has been the subject of many papers and we can make use of such previously published results. We will make use of the formulae of van Oosterom [96], which are compact and were explicitly derived with stability concerns in mind.

Let $F \subset \mathbb{R}^3$ be a non-degenerate, planar triangle in space. We want to evaluate the following integral for a linearly varying function $\boldsymbol{\sigma}$:

$$\frac{1}{4\pi} \int_F \frac{\boldsymbol{\sigma}}{r} \,\mathrm{d}S(\mathbf{y}). \tag{5.9}$$

Let $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$ denote the corners of F and for notational convenience set $\mathbf{y}_4 := \mathbf{y}_1$ and $\mathbf{y}_5 := \mathbf{y}_2$. Let $e_i = |\mathbf{y}_{i+1} - \mathbf{y}_i|$ denote the lengths of the triangle's edges and let $\mathbf{e}_i = (\mathbf{y}_{i+1} - \mathbf{y}_i)/e_i$ denote its normalised edge-vectors. Furthermore let $\boldsymbol{\sigma}_i$ denote the value of $\boldsymbol{\sigma}$ at node $\mathbf{y}_i$, let $|F|$ denote the area of F and let $\mathbf{n}$ denote its unit normal vector, pointing in the direction of $\mathbf{e}_i \times \mathbf{e}_{i+1}$. Finally, letting $\mathbf{r}_i = \mathbf{y}_i - \mathbf{x}$, $r_i = |\mathbf{r}_i|$ as before, we have:

$$\frac{1}{4\pi} \int_F \frac{\boldsymbol{\sigma}}{r} \,\mathrm{d}S(\mathbf{y}) = \sum_{i=1}^{3} \frac{\boldsymbol{\sigma}_i}{4\pi} \frac{e_{i+1}}{2|F|} \Big([\mathbf{r}_{i+1}, \mathbf{e}_{i+1}, \mathbf{n}] A - \mathbf{e}_{i+1} \cdot \mathbf{B} \Big). \tag{5.10}$$

Here $[\cdot,\cdot,\cdot]$ denotes the scalar triple-product and:

$$A = \left(\sum_{i=1}^{3} [\mathbf{r}_i, \mathbf{e}_i, \mathbf{n}] L_i\right) - d\Omega_F, \quad \mathbf{B} = \sum_{i=1}^{3} M_i \mathbf{e}_i, \tag{5.11}$$

$$\Omega_F = 2\,\mathrm{atan2}\big([\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3],\ r_1 r_2 r_3 + r_1(\mathbf{r}_2 \cdot \mathbf{r}_3) + r_2(\mathbf{r}_1 \cdot \mathbf{r}_3) + r_3(\mathbf{r}_1 \cdot \mathbf{r}_2)\big). \tag{5.12}$$

Ω_F is the solid angle of the triangle F subtended at $\mathbf{x}$ and $d := \mathbf{r} \cdot \mathbf{n}$ is the signed distance of the point $\mathbf{x}$ to the plane of F. atan2 refers to the arctangent function with two arguments. Note that as $\mathbf{x}$ approaches the plane of F, d tends to zero. As long as atan2 is defined to take finite values if both of its arguments are zero, these expressions are numerically stable. This is the case for its commonly used definition from IEEE 754 [97].

The numbers L_i and M_i correspond to edge integrals which are independent of $\boldsymbol{\sigma}$, $\mathbf{n}$, and the orientation of the edge. Thus, their values need to be computed only once for each edge and can be reused for the other face of T sharing this edge. They are given by:

$$M_i = \frac{r_i + r_{i+1}}{4e_i}\left[e_i^2 + (r_{i+1} - r_i)^2\right] + \tfrac{1}{2}|\mathbf{r}_i \times \mathbf{e}_i|^2 L_i, \tag{5.13}$$

$$L_i = \ln \frac{(r_{i+1} + e_i)^2 - r_i^2}{r_{i+1}^2 - (r_i - e_i)^2}. \tag{5.14}$$

The expression for L_i becomes numerically unstable when the field point $\mathbf{x}$ approaches the line the edge lies on. As van Oosterom points out and demonstrates with numerical experiments in his paper [96], a simple, numerically accurate, and efficient way to avoid this problem in a computer implementation is to replace it with:

$$L_i = \ln \frac{(r_{i+1} + e_i)^2 - r_i^2 + \varepsilon}{r_{i+1}^2 - (r_i - e_i)^2 + \varepsilon}, \tag{5.15}$$

where ε denotes the machine epsilon.

5.4 Numerical Example and Conclusion

In order to illustrate the benefits of the new formulation, as an example we take the tetrahedron defined by the nodes $\mathbf{y}_1 = \mathbf{0}$, $\mathbf{y}_2 = \hat{\boldsymbol{\imath}}$, $\mathbf{y}_3 = \hat{\boldsymbol{\jmath}}$, $\mathbf{y}_4 = \hat{\boldsymbol{k}}$, let $\boldsymbol{\omega}_1 = (1, 0, 0)$, and set the remaining $\boldsymbol{\omega}_i = \mathbf{0}$. In figure 5.1 we plot the absolute value of the resulting velocity along two parallel lines with a distance of one machine epsilon. The computations were performed using double precision floating-point arithmetic. In the first case the line runs along one of the tetrahedron's faces and both methods produce identical results. The second line runs through the tetrahedron's inside, but very close to the same face. Numerical difficulties in determining the correct value of α cause seemingly unpredictable errors in the method of Suh, while the formula of van Oosterom remain accurate and yield the correct result.

We conclude that the presented formula are simpler, cheaper to evaluate, and numerically more stable than the previously published approach.

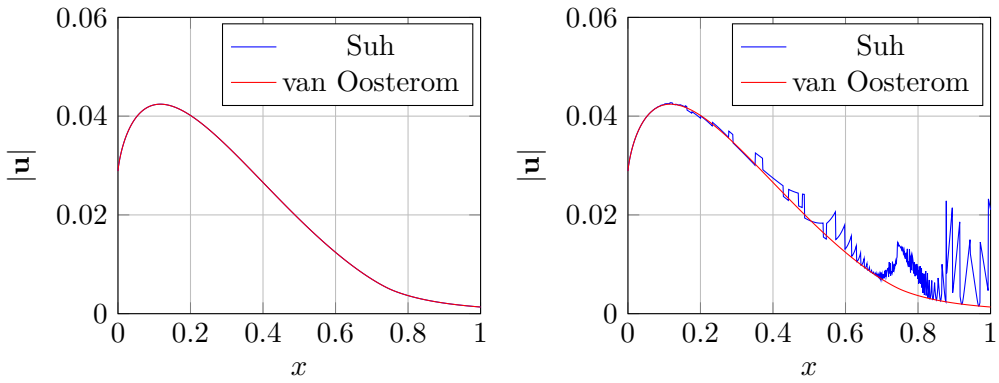

Figure 5.1: On the left is a plot of the velocity magnitude along the axis $(x, 0.25, 0)$. The results of both methods coincide. On the right is a plot along the axis $(x, 0.25, \varepsilon)$. Difficulties in determining the correct value of α cause the results of Suh's method to jump seemingly unpredictably.

Chapter 6

Conclusions and Outlook

This work was originally driven by the desire to extend vortex methods to viscous flows in bounded domains, while at the same time maintaining their mesh-free character. After some further investigations into the theory of these methods, this turned out to be a mammoth task. Fortunately, several aspects of this task can be considered in isolation. In this work we looked at some of these aspects and proposed new approaches to their solution. In this chapter we present our conclusions and give an outlook for possible future research.

6.1 Conclusions

In chapter 3 we introduced a viscous scheme for the whole-space case which requires neither frequent remeshing nor the viscous splitting. The splitting-free vorticity redistribution scheme shows that one can apply truly mesh-free particle methods to such flows.

The treatment of bounded flows, however, turned out to be a significantly more difficult topic—even in the inviscid case. In vortex methods all we are given is a quadrature rule $\boldsymbol{\omega}_h$ for integrating smooth functions against the underlying vorticity we are trying to approximate. In chapter 4 we proposed a new scheme for the problem of particle regularisation and proved its optimality and stability. This approach has all the benefits of the conventionally used on blob functions; in particular it conserves all moments up to a user defined order. Due to its formulation as a stabilised fictitious domain method, this approach works for a very general class of bounded domains. To the best of our knowledge, this is the first regularisation scheme that works for such a general class of domains while only requiring quadrature rules to be available. The scheme maps particle approximations $\boldsymbol{\omega}_h$ to smooth functions $\boldsymbol{\omega}_\sigma$ and tells us to choose σ proportional to $\sqrt{h}$. As a direct consequence the velocity evaluation could be drastically sped up. We believe that this scheme finally opens the door for the application of vortex methods in domains with boundaries, and we discuss some lines of possible future research in the following section.

6.2 Outlook

As an alternative to the evaluation the Biot–Savart law using the formulae from chapter 5, one could also consider inverting the kinematic system (2.11) by using mesh based Poisson solvers on the equation $-\Delta\boldsymbol{\Psi} = \boldsymbol{\omega}$. They have the advantage of being

very fast, however, typically cannot handle the far-field condition accurately. On the other hand, such solvers could again use a fictitious domain formulation and work on $\boldsymbol{\omega}_\sigma$ directly. Boundary conditions could for example be enforced weakly with the help of Nitsche's method [98].

Viscosity has a mollifying effect on the vorticity field, and can thus only further increase its length scales. It thus makes sense to hybridise the method and solve the viscous effects using $\boldsymbol{\omega}_\sigma$ instead of $\boldsymbol{\omega}_h$. This could for example be done using a mesh based solver, which are well known to handle viscous effects and boundary conditions accurately. A natural approach would be to again use a fictitious domain formulation. One then furthermore requires a scheme that maps the resulting smooth functions $\boldsymbol{\omega}_\sigma$ back to the particle field, to handle convection and stretching.

A simple approach to obtain a particle approximation from such a smooth function would be to reinitialise the particle field using the available quadrature rules. This would correspond to a remeshing at every time step and effectively destroy the Lagrangian character of vortex methods. Instead one could try to compute quadrature rules for the domain using the particle locations as prescribed nodes. This amounts to determining new quadrature weights w_i. Strain successfully used such an approach for the two-dimensional inviscid equations [99]. Additionally enforcing positivity of the weights yields phase I problems with Vandermonde matrices that closely resemble the ones encountered in chapter 3.

What also remains to be answered is how to couple mesh and particle based solvers in a time-stepping scheme and how to treat the no-slip condition. A popular approach [76, 100, 101] is to first prescribe homogeneous Neumann conditions on the vorticity and advance the solution one step in time. This boundary condition is usually incorrect and the vorticity field becomes inadmissible. In a second step one then corrects this error by diffusing the resulting 'spurious vortex sheet', which effectively corresponds to the difference of $\mathbf{u}_{\partial D}$ and $\mathbf{u}_\varphi$ as described in subsection 2.2.3. This approach can be seen as a discretised version of the continuous algorithm by Cottet and Koumoutsakos [48, Section 6.3.3]. It represents a viscous splitting for the vorticity creation at the boundaries and thus limits the time accuracy of the method.

An interesting topic for future research would thus be the question how vorticity boundary conditions can be handled with more advanced time-stepping schemes and a hybrid vortex method. Of particular interest are the IMEX methods [78]. These time-stepping schemes handle the viscous, stiff part of the equation using an implicit scheme and use explicit methods for the remaining non-stiff parts, while still achieving high order accuracy in time. Such a scheme could potentially combine the best parts of both worlds: the implicit time discretisation combined with a mesh based solver for the viscous term could accurately handle diffusive effects and boundary conditions without restrictive time-step constraints. The convective parts of the equation would be solved using explicit methods on the particle field and benefit from he resulting lack of artificial viscosity. Whether or not it is possible to implement such a method in practice remains to be seen.

Appendix A

Sobolev Spaces

The analysis of the methods presented in this book are based on the notion of Sobolev spaces. The theory of these spaces is too vast to be treated in detail here, and the unfamiliar reader is referred to the literature, for example to the book of Adams and Fournier [68]. They especially describe and prove several embedding properties of these spaces, which are of importance in this thesis. There are several differing definitions of these spaces, which often—but not always—can be shown to be equivalent. In order to avoid confusion in these cases, we will briefly define the necessary spaces encountered throughout this thesis.

Under the term *domain* we understand an open, connected subset D of $\mathbb{R}^d$, $d \in \{2, 3\}$. By subset we will always refer to the weaker notion $D \subseteq \mathbb{R}^d$, such that D may equal $\mathbb{R}^d$. We will always use the symbol '$\subset$' instead of '$\subseteq$'; for a proper subset we will write $D \subsetneq \mathbb{R}^d$. We will always assume that such domains are reasonably well shaped, such that we can apply the Sobolev embedding theorem and other important results from analysis. This is for example the case for Lipschitz domains, and it can essentially be always assumed for domains from engineering applications. For such domains we can define spaces of the Lebesgue integrable functions $L^p(D)$, for $p \in [1, \infty]$:

$$L^p(D) := \{f : D \to \mathbb{R} \,|\, \|f\|_{L^p(D)} < \infty\}, \tag{A.1}$$

where the corresponding Lebesgue norms are defined as:

$$\|f\|_{L^p(D)} := \begin{cases} \left(\int_D |f|^p \,\mathrm{d}V\right)^{1/p} & \text{if } p \in [1, \infty), \\ \operatorname{ess\,sup}_{x \in D} |f(x)| & \text{if } p = \infty. \end{cases} \tag{A.2}$$

We will consider two functions $f, g \in L^p(D)$ to be equivalent if $\|f - g\|_{L^p(D)} = 0$. This means that f and g may only differ on a subset of D that has zero measure. In this case we will write $f = g$. Thus, strictly speaking, the $L^p(D)$ spaces consist of equivalence classes of functions, which only differ on a subset of D of measure zero. For brevity we will write L^p instead of $L^p(D)$ if the domain in question is clear. All of these spaces are Banach spaces; $L^2(D)$ is a Hilbert space under the scalar product $(f, g)_{L^2(D)} := \int_D fg \,\mathrm{d}x$.

For $k \in \mathbb{N}_0$ we define $C^k(D)$ to be the space of k times continuously differentiable functions on D, with the additional property that the following norm is finite:

$$\|f\|_{C^k(D)} := \max_{|\alpha| \leq k} \sup_{x \in D} |D^\alpha f(x)|. \tag{A.3}$$

Here, α is a multi-index and the symbol D^α refers to the conventional, strong derivative. This norm is also called the supremum norm. Equipped with this norm, the spaces $C^k(D)$ are Banach spaces. We will simply write $C(D)$ for the space of continuous functions, i. e., the case $k = 0$.

The support of a function $f : D \to \mathbb{R}$ is defined to be the closure of the set of those $x \in D$ for which f does not vanish:

$$\operatorname{supp} f := \operatorname{clos}\{x \in D \,|\, f(x) \neq 0\}. \tag{A.4}$$

We say a function f is compactly supported if its support is compact and a subset of the domain: $\operatorname{supp} f \subset\subset D$. In this case we will write $f \in C_0^k(D)$. Because D is open and a compact set is always closed, this automatically means that $f \in C_0^k(D) \Longrightarrow \operatorname{supp} f \subsetneq D$. This especially implies that f and all of its derivatives vanish at the domain's boundary ∂D. Equipped with the supremum norm as defined above, the spaces $C_0^k(D)$ are also Banach spaces.

Let $f \in L^p(D)$ and let α be a multi-index. If there exists a function $g \in L^p(D)$ such that:

$$\int_D g\varphi \,\mathrm{d}x = (-1)^{|\alpha|} \int_D f\partial^\alpha \varphi \,\mathrm{d}x \qquad \forall \varphi \in C_0^\infty(D), \tag{A.5}$$

then g is called the αth weak derivative of f and we write $\partial^\alpha f := g$. If a weak derivative exists it is unique. Furthermore, if f is also differentiable in the strong, conventional sense, the weak and strong derivatives coincide. With the help of this, we can define the Sobolev spaces $W^{k,p}(D)$, $k \in \mathbb{N}$, $p \in [1, \infty]$:

$$W^{k,p}(D) := \{f \in L^p(D) \,|\, \|f\|_{W^{k,p}(D)} < \infty\}, \tag{A.6}$$

where the Sobolev norms are defined by:

$$\|f\|_{W^{k,p}(D)} := \begin{cases} (\sum_{|\alpha| \leq k} \|\partial^\alpha f\|_{L^p(D)}^p)^{1/p} & \text{if } p \in [1, \infty), \\ \max_{|\alpha| \leq k} \|\partial^\alpha f\|_{L^\infty(D)} & \text{if } p = \infty. \end{cases} \tag{A.7}$$

Additionally, we set $W^{0,p}(D) := L^p(D)$, $p \in [1, \infty]$. All of these spaces are Banach spaces. Under the scalar product $(f, g)_{W^{k,2}(D)} := \sum_{|\alpha| \leq k} (\partial^\alpha f, \partial^\alpha g)_{L^2(D)}$ the spaces $W^{k,2}(D)$ become Hilbert spaces. We briefly remark that due to the Meyers–Serrin theorem [102], many of the various definitions of the Sobolev spaces found in the literature yield the same spaces, with the important exception of the case $p = \infty$. Analogously to the space $C_0^k(D)$ one can define $W_0^{k,p}(D)$ to be the space of those functions $f \in W^{k,p}(D)$, that have a compact support and vanish at the boundary ∂D. One can furthermore show that $W_0^{k,p}(\mathbb{R}^d) = W^{k,p}(\mathbb{R}^d)$.

In the study of numerical schemes it is often useful to look at the norm of the highest order derivatives of the functions in question. To this end we also introduce the Sobolev semi-norms:

$$|f|_{W^{k,p}(D)} := \begin{cases} (\sum_{|\alpha| = k} \|\partial^\alpha f\|_{L^p(D)}^p)^{1/p} & \text{if } p \in [1, \infty), \\ \max_{|\alpha| = k} \|\partial^\alpha f\|_{L^\infty(D)} & \text{if } p = \infty. \end{cases} \tag{A.8}$$

Particle approximations ω_h can best be understood to be approximations for integrating a given sufficiently smooth function against an underlying function ω that we are trying to approximate. Thus, a particle approximation maps a given smooth function to a real number. A mapping that maps an element of a vector space to a scalar is commonly called a functional. The space of all bounded functionals on a vector space is called the dual space.

Let $W^{k,p}(D)$, $k \in \mathbb{N}$, $k \neq 0$, $p \in [1, \infty]$ be a given Sobolev space. We then *define* $W^{-k,q}(D)$ to be its dual space, where q is the so-called Hölder conjugate to p. It is defined such that for a given p we have $1 = 1/p + 1/q$; $q = 1$ if $p = \infty$. This *definition* of negative index Sobolev spaces is in accordance with the one used by Scott and Brenner [60, Chapter 1]. Note that, for example, Adams and Fournier define the negative index spaces differently [68]; the resulting spaces usually *do not coincide with the spaces defined here.*

Dual spaces spaces come with a natural norm. For a functional $f \in W^{-k,q}(D)$ this norm is defined as:

$$\|f\|_{W^{-k,q}(D)} := \sup_{\varphi \in W^{k,p}(D)} \frac{|\langle f, \varphi \rangle|}{\|\varphi\|_{W^{k,p}(D)}}. \tag{A.9}$$

The notation $\langle f, \varphi \rangle$ refers to the application of the functional f to the function φ; another way of writing this is $f(\varphi)$. Under this norm the spaces $W^{-k,q}(D)$ become Banach spaces. The symbol $\langle \cdot, \cdot \rangle$ is called the dual pairing. The definition of this norm is such that for any function $\varphi \in W^{k,p}(D)$ we always have $|f(\varphi)| \leq \|f\|_{W^{-k,q}(D)} \|\varphi\|_{W^{k,p}(D)}$.

We remark that for any $f \in L^q(D)$ we can define a corresponding functional on $W^{k,p}(D)$, $k \in \mathbb{N}$, $p \in [1, \infty]$:

$$\langle f, \varphi \rangle := \int_D f \varphi \, \mathrm{d}x, \qquad \varphi \in W^{k,p}(D), f \in L^q(D). \tag{A.10}$$

In this way, the notation $\langle \cdot, \cdot \rangle$ for the dual paring is reminiscent of the $L^2(D)$ scalar product. However, the converse is generally not true: not every functional $f \in W^{-k,q}$ has a corresponding function in $L^q(D)$ such that it can be written in the above form. This is for example the case for the Dirac delta functional δ, which evaluates a given function at the origin: $\langle \delta, \varphi \rangle := \varphi(0)$.

The Dirac delta functional is maybe the most important notion in the field of vortex methods. In order for this functional to be well defined, we will need to assume that φ is at least continuous on a domain D that contains the origin, i. e., $\varphi \in C(D)$. Under certain mild conditions on the boundary ∂D of the domain, the Sobolev embedding theorem guarantees that this can for example always be assumed if $\varphi \in W^{k,p}(D)$, $k > d/p$, $p \in [1, \infty]$, or if $k \geq d$ if $p = 1$. In this case there exists a unique continuous function in the equivalence class of φ. The Dirac delta functional is then defined to evaluate this continuous function at the origin, and one can show that $\|\delta\|_{W^{-k,q}} \leq C_{\mathrm{emb}}$, with the Sobolev embedding constant C_{emb}, which may depend on the domain as well as k, d, and p.

Bibliography

[1] H. L. F. von Helmholtz. 'Über Integrale der hydrodynamischen Gleichungen, welche den Wirbelbewegungen entsprechen'. In: *Journal für die reine und angewandte Mathematik* 55 (1858), pp. 25–55.

[2] L. Euler. 'Principes généraux de l'état d'équilibre des fluides'. In: *Histoire de l'Académie Royale des Sciences et des Belles-Lettres de Berlin* (1757): *Année MDCCLV*, pp. 217–273.

[3] G. G. Stokes. 'On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solids'. In: *Transactions of the Cambridge Philosophical Society* 8 (1845), pp. 287–319.

[4] F. Klein. 'Über die Bildung von Wirbeln in reibungslosen Flüssigkeiten'. In: *Zeitschrift für Mathematik und Physik. Organ für Angewandte Mathematik* 58 (1910), pp. 259–262.

[5] H. L. F. von Helmholtz. 'Discontinuirliche Flüssigkeits-Bewegungen'. In: *Monatsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin* (1869): *Aus dem Jahre 1868*, pp. 215–228.

[6] W. Thomson (1st Baron Kelvin). 'Hydrokinetic Solutions and Observations'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 4th ser. vol. 42 (Nov. 1871), pp. 362–377. ISSN: 1478–6435.

[7] J. W. Strutt (3rd Baron Rayleigh). 'On the Instability of Jets'. In: *Proceedings of the London Mathematical Society* 10.1 (Nov. 1878), pp. 4–13. ISSN: 0024–6115. DOI: 10.1112/plms/s1-10.1.4.

[8] W. Thomson (1st Baron Kelvin). 'On the doctrine of discontinuity of fluid motion, in connection with the resistance against a solid moving through a fluid'. In: *Nature* 50 (1894), pp. 524f, 549f, 573f, 579f. ISSN: 0028–0836.

[9] L. Rosenhead. 'The Formation of Vortices from a Surface of Discontinuity'. In: *Proceedings of the Royal Society of London* 142.832 (Nov. 1931), pp. 170–192.

[10] G. Birkhoff and J. Fisher. 'Do vortex sheets roll up?' In: *Rendiconti del Circolo Matematico di Palermo* 8.1 (Jan. 1959), pp. 77–90. ISSN: 0009–725X. DOI: 10.1007/BF02843773.

[11] R. Krasny. 'A study of singularity formation in a vortex sheet by the point-vortex approximation'. In: *Journal of Fluid Mechanics* 167 (June 1986), pp. 65–93. ISSN: 0022-1120. DOI: 10.1017/S0022112086002732.

[12] F. H. Harlow. 'Hydrodynamic Problems Involving Large Fluid Distortions'. In: *Journal of the Association of Computing Machinery* 4.2 (Apr. 1957), pp. 137–142. ISSN: 0004–5411. DOI: 10.1145/320868.320871.

[13] M. W. Evans and F. H. Harlow. *The Particle-in-Cell Method for Hydrodynamic Calculations*. Report LA-2139. Los Alamos Scientific Laboratory of the University of California, 1957.

[14] C. K. Birdsall and D. Fuss. 'Clouds-in-Clouds, Clouds-in-Cells Physics for Many-Body Plasma Simulation'. In: *Journal of Computational Physics* 3.4 (1969), pp. 494–511. ISSN: 0021–9991. DOI: 10.1016/0021-9991(69)90058-8.

[15] V. V. Meleshko and H. Aref. 'A Bibliography of Vortex Dynamics 1858–1956'. In: *Advances in Applied Mechanics*. Ed. by H. Aref and E. van der Giessen. Vol. 41. Elsevier, 2007, pp. 197–292. ISBN: 9780120020577. DOI: 10.1016/S0065-2156(07)41003-1.

[16] L. Prandtl. 'Über Flüssigkeitsbewegung bei sehr kleiner Reibung'. In: *Verhandlung-en des III. Mathematiker-Kongresses, Heidelberg 1904*. Leipzig: Teubner, 1905, pp. 484–491.

[17] A. J. Majda and A. L. Bertozzi. *Vorticity and Incompressible Flow*. Cambridge University Press, Nov. 2001. ISBN: 0521630576.

[18] R. Krasny. 'Desingularization of periodic vortex sheet roll-up'. In: *Journal of Computational Physics* 65.2 (Aug. 1986), pp. 292–313. ISSN: 0021–9991. DOI: 10.1016/0021-9991(86)90210-X.

[19] J. P. Christiansen. 'Numerical Simulation of Hydrodynamics by the Method of Point Vortices'. In: *Journal of Computational Physics* 13.3 (Nov. 1973), pp. 363–379. ISSN: 0021–9991. DOI: 10.1016/0021-9991(73)90042-9.

[20] A. J. Chorin. 'Numerical study of slightly viscous flow'. In: *Journal of Fluid Mechanics* 57.4 (Mar. 1973), pp. 785–796. ISSN: 0022–1120. DOI: 10.1017/S0022112073002016.

[21] A. J. Chorin and P. S. Bernard. 'Discretization of a vortex sheet, with an example of roll-up'. In: *Journal of Computational Physics* 13.3 (Nov. 1973), pp. 423–429. ISSN: 0021–9991. DOI: 10.1016/0021-9991(73)90045-4.

[22] F. Milinazzo and P. G. Saffman. 'The Calculation of Large Reynolds Number Two-Dimensional Flow Using Discrete Vortices with Random Walk'. In: *Journal of Computational Physics* 23.4 (Apr. 1977), pp. 380–392. ISSN: 0021–9991. DOI: 10.1016/0021-9991(77)90069-9.

[23] T. E. Dushane. 'Convergence for a Vortex Method for Solving Euler's Equation'. In: *Mathematics of Computation* 27.124 (Oct. 1973), pp. 719–728. ISSN: 0025–5718. DOI: 10.2307/2005505.

[24] O. H. Hald and V. Mauceri Del Prete. 'Convergence of Vortex Methods for Euler's Equations'. In: *Mathematics of Computation* 32.143 (July 1978), pp. 791–809. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-1978-0492039-1.

[25] O. H. Hald. 'Convergence of Vortex Methods for Euler's Equations. II'. In: *SIAM Journal on Numerical Analysis* 16.5 (Oct. 1979), pp. 726–755. ISSN: 0036–1429. DOI: 10.1137/0716055.

[26] J. T. Beale and A. J. Majda. 'Vortex Methods. I. Convergence in Three Dimensions'. In: *Mathematics of Computation* 39.159 (July 1982), pp. 1–27. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-1982-0658212-5.

[27] J. T. Beale and A. J. Majda. 'Vortex Methods. II: Higher Order Accuracy in Two and Three Dimensions'. In: *Mathematics of Computation* 39.159 (July 1982), pp. 29–52. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-1982-0658213-7.

[28] J. T. Beale and A. J. Majda. 'High Order Accurate Vortex Methods with Explicit Velocity Kernels'. In: *Journal of Computational Physics* 58.2 (Apr. 1985), pp. 188–208. ISSN: 0021–9991. DOI: 10.1016/0021-9991(85)90176-7.

[29] J. T. Beale. 'A Convergent 3-D Vortex Method With Grid-Free Stretching'. In: *Mathematics of Computation* 46.174 (Apr. 1986), pp. 401–424. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-1986-0829616-6.

[30] O. H. Hald. 'Convergence of Vortex Methods for Euler's Equations, III'. In: *SIAM Journal on Numerical Analysis* 24.3 (May 1987), pp. 538–582. ISSN: 0036–1429. DOI: 10.1137/0724039.

[31] C. Anderson and C. Greengard. 'On Vortex Methods'. In: *SIAM Journal on Numerical Analysis* 22.3 (June 1985), pp. 413–440. ISSN: 0036–1429. DOI: 10.1137/0722025.

[32] J. T. Beale and A. J. Majda. 'Rates of convergence for viscous splitting of the Navier–Stokes equations'. In: *Mathematics of Computation* 37.156 (Oct. 1981), pp. 243–259. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-1981-0628693-0.

[33] J. Goodman. 'Convergence of the Random Vortex Method'. In: *Communications on Pure and Applied Mathematics* 40.2 (Mar. 1987), pp. 189–220. ISSN: 1097–0312. DOI: 10.1002/cpa.3160400204.

[34] D.-G. Long. 'Convergence of the Random Vortex Method in Two Dimensions'. In: *Journal of the American Mathematical Society* 1.4 (Oct. 1988), pp. 779–804. ISSN: 0894–0347. DOI: 10.1090/S0894-0347-1988-0958446-1.

[35] G. R. Baker. 'The "Cloud in Cell" Technique Applied to the Roll Up of Vortex Sheets'. In: *Journal of Computational Physics* 31.1 (Apr. 1979), pp. 76–95. ISSN: 0021–9991. DOI: 10.1016/0021-9991(79)90063-9.

[36] R. W. Hockney, S. P. Goel, and J. W. Eastwood. 'Quiet High-Resolution Computer Models of a Plasma'. In: *Journal of Computational Physics* 14.2 (Feb. 1974), pp. 148–158. ISSN: 0021–9991. DOI: 10.1016/0021-9991(74)90010-2.

[37] B. Couët, O. Buneman, and A. Leonard. 'Simulation of Three-Dimensional Incompressible Flows with a Vortex-in-Cell Method'. In: *Journal of Computational Physics* 39.2 (Feb. 1981), pp. 305–328. ISSN: 0021–9991. DOI: 10.1016/0021-9991(81)90154-6.

[38] A. Leonard. 'Vortex Methods for Flow Simulation'. In: *Journal of Computational Physics* 37.3 (Oct. 1980), pp. 289–335. ISSN: 0021–9991. DOI: 10.1016/0021-9991(80)90040-6.

[39] A. Leonard. 'Computing Three-Dimensional Incompressible Flows with Vortex Elements'. In: *Annual Reviews of Fluid Mechanics* 17 (1985), pp. 523–559. ISSN: 0066–4189. DOI: 10.1146/annurev.fl.17.010185.002515.

[40] R. A. Gingold and J. J. Monaghan. 'Smoothed particle hydrodynamics: theory and application to non-spherical stars'. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389. ISSN: 0035–8711. DOI: 10.1093/mnras/181.3.375.

[41] L. B. Lucy. 'A numerical approach to the testing of the fission hypothesis'. In: *The Astronomical Journal* 82.12 (Dec. 1977), pp. 1013–1024. ISSN: 0004–6256. DOI: 10.1086/112164.

[42] J. Barnes and P. Hut. 'A hierarchical O(N log N) force-calculation algorithm'. In: *Nature* 324 (Dec. 1986), pp. 446–449. ISSN: 0028–0836. DOI: 10.1038/324446a0.

[43] L. F. Greengard and V. Rokhlin. 'A fast algorithm for particle simulations'. In: *Journal of Computational Physics* 73.2 (Dec. 1987), pp. 325–348. ISSN: 0021–9991. DOI: 10.1016/0021-9991(87)90140-9.

[44] W. Dehnen. 'A Hierarchical O(N) Force Calculation Algorithm'. In: *Journal of Computational Physics* 179.1 (June 2002), pp. 27–42. ISSN: 0021–9991. DOI: 10.1006/jcph.2002.7026.

[45] P.-A. Raviart. 'An Analysis of Particle Methods'. In: *Numerical Methods in Fluid Dynamics*. Ed. by F. Brezzi. Vol. 1127. Lecture Notes in Mathematics. Springer, 1984. Chap. 4, pp. 243–324. ISBN: 9783540152255. DOI: 10.1007/BFb0074532.

[46] G.-H. Cottet. 'A new approach for the analysis of Vortex Methods in two and three dimensions'. In: *Annales de l'Institut Henri Poincaré. Analyse non linéaire* 5.3 (1988), pp. 227–285. ISSN: 0294–1449. DOI: 10.1016/S0294-1449(16)30346-8.

[47] R. Yokota, L. A. Barba, T. Narumi, and K. Yasuoka. 'Petascale turbulence simulation using a highly parallel fast multipole method on GPUs'. In: *Computer Physics Communications* 184.3 (Mar. 2013), pp. 445–455. ISSN: 0010–4655. DOI: 10.1016/j.cpc.2012.09.011.

[48] G.-H. Cottet and P. D. Koumoutsakos. *Vortex Methods. Theory and Practice*. Cambridge University Press, 2000. ISBN: 0521621860.

[49] P. Koumoutsakos. 'Multiscale Flow Simulations Using Particles'. In: *Annual Reviews of Fluid Mechanics* 37 (2005), pp. 457–487. ISSN: 0066–4189. DOI: 10.1146/annurev.fluid.37.061903.175753.

[50] J. Kurzak and B. M. Pettitt. 'Fast Multipole Methods for Particle Dynamics'. In: *Molecular Simulation* 23.10–11 (2006), pp. 775–790. ISSN: 0892–7022. DOI: 10.1080/08927020600991161.

[51] S. A. Sauter and C. Schwab. *Boundary Element Methods*. Springer Series in Computational Mathematics 39. Springer, 2011. ISBN: 9783540680925.

[52] H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer. 'The Helmholtz–Hodge Decompostion—A Survey'. In: *IEEE Transactions on Visualization and Computer Graphics* 19.8 (Aug. 2013), pp. 1386–1404. ISSN: 1077–2626. DOI: 10.1109/TVCG.2012.316.

[53] A. P. Calderón and A. Zygmund. 'On the Existence of Certain Singular Integrals'. In: *Acta Mathematica* 88 (1952), pp. 85–139. ISSN: 0001–5962. DOI: 10.1007/BF02392130.

[54] E. M. Stein. *Singular Integrals and Differentiability Properties of Functions*. Vol. 30. Princeton Mathematical Series. Princeton University Press, 1970. ISBN: 0691080798.

[55] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*. 2nd ed. Vol. 25. Springer Series in Computational Mathematics. Springer, 2006. ISBN: 3540331212.

[56] G. H. Golub and J. H. Welsch. 'Calculation of Gauss quadrature rules'. In: *Mathematics of Computation* 23.106 (1969), pp. 221–230. ISSN: 0025–5718. DOI: 10.1090/S0025-5718-69-99647-1.

[57] L. Zhang, T. Cui, and H. Liu. 'A Set of Symmetric Quadrature Rules on Triangles and Tetrahedra'. In: *Journal of Computational Mathematics* 27.1 (Jan. 2009), pp. 89–96. ISSN: 0254–9409.

[58] C. Geuzaine and J.-F. Remacle. 'Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities'. In: *International Journal for Numerical Methods in Engineering* 79.11 (Sept. 2009), pp. 1309–1331. ISSN: 1097–0207. DOI: 10.1002/nme.2579.

[59] H. Si. 'TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator'. In: *ACM Transactions on Mathematical Software* 41.2 (Jan. 2015), 11:1–36. ISSN: 0098–3500. DOI: 10.1145/2629697.

[60] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. 3rd ed. Vol. 15. Texts in Applied Mathematics. Springer, 2008. ISBN: 9780387759333. DOI: 10.1007/978-0-387-75934-0.

[61] L. A. Barba. 'Vortex Method for computing high-Reynolds number flows. Increased accuracy with a fully mesh-less formulation'. PhD thesis. California Institute of Technology, May 2004.

[62] C. Greengard. 'The core spreading method approximates the wrong equation'. In: *Journal of Computational Physics* 61.2 (Nov. 1985), pp. 345–348. ISSN: 0021–9991. DOI: 10.1016/0021-9991(85)90091-9.

[63] L. F. Rossi. 'Resurrecting Core Spreading Vortex Methods: A New Scheme that is Both Deterministic and Convergent'. In: *SIAM Journal on Scientific Computing* 17.2 (1996), pp. 370–397. ISSN: 1064–8275. DOI: 10.1137/S1064827593254397.

[64] S. Shankar and L. van Dommelen. 'A New Diffusion Procedure for Vortex Methods'. In: *Journal of Computational Physics* 127.1 (Aug. 1996), pp. 88–109. ISSN: 0021–9991. DOI: 10.1006/jcph.1996.0160.

[65] B. Schrader, S. Reboux, and I. F. Sbalzarini. 'Discretization correction of general integral PSE Operators for particle methods'. In: *Journal of Computational Phsyics* 229.11 (June 2010), pp. 4159–4182. ISSN: 0021–9991. DOI: 10.1016/j.jcp.2010.02.004.

[66] B. Seibold. 'M-Matrices in Meshless Finite Difference Methods'. PhD thesis. Technische Universität Kaiserslautern, Sept. 2006.

[67] B. Seibold. 'Minimal positive stencils in meshfree finite difference methods for the Poisson equation'. In: *Computer Methods in Applied Mechanics and Engineering* 198.3–4 (Dec. 2008), pp. 592–601. ISSN: 0045–7825. DOI: 10.1016/j.cma.2008.09.001.

[68] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. 2nd ed. Pure and Applied Mathematics 140. Elsevier, 2003. ISBN: 9780120441433.

[69] B. Seibold. 'Performance of algebraic multigrid methods for non-symmetric matrices arising in particle methods'. In: *Numerical Linear Algebra with Applications* 17.2–3 (Apr. 2010), pp. 433–451. ISSN: 1099–1506. DOI: 10.1002/nla.710.

[70] S. Gottlieb, C.-W. Shu, and E. Tadmor. 'Strong Stability-Preserving High-Order Time Discretization Methods'. In: *SIAM Review* 43.1 (2001), pp. 89–112. ISSN: 0036–1445. DOI: 10.1137/S003614450036757X.

[71] T. Tao. *Conserved quantities for the Euler equations*. Personal Blog Entry. Feb. 2014. URL: https://terrytao.wordpress.com.

[72] R. Fletcher. *Practical Methods of Optimization*. 2nd ed. Wiley, July 2000. ISBN: 9780471494638.

[73] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th ed. John Hopkins University Press, 2013. ISBN: 1421407949.

[74] E. Anderson et al. *LAPACK User's Guide*. Society for Industrial and Applied Mathematics, 1999. ISBN: 0898714478.

[75] R. Fletcher and S. P. J. Matthews. 'Stable modification of explicit LU factors for simplex updates'. In: *Mathematical Programming* 30.3 (Oct. 1984), pp. 267–284. ISSN: 0025–5610. DOI: 10.1007/BF02591933.

[76] I. Lakkis and A. Ghoniem. 'A high resolution spatially adaptive vortex method for separating flows. Part I: Two-dimensional domains'. In: *Journal of Computational Physics* 228.2 (Feb. 2009), pp. 491–515. ISSN: 0021–9991. DOI: 10.1016/j.jcp.2008.09.025.

[77] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Classics in Applied Mathematics 15. Society for Industrial and Applied Mathematics, 1995. ISBN: 9780898713565.

[78] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. 'Implicit-Explicit Methods for Time-Dependent Partial Differential Equations'. In: *SIAM Journal on Numerical Analysis* 32.3 (June 1995), pp. 797–823. ISSN: 0036–1429. DOI: 10.1137/0732037.

[79] Z.-H. Teng. 'Elliptic-vortex method for incompressible flow at high Reynolds Number'. In: *Journal of Computational Phsyics* 41.1 (Aug. 1982), pp. 54–68. ISSN: 0021–9991. DOI: 10.1016/0021-9991(82)90005-5.

[80] G. Russo and J. A. Strain. 'Fast Triangulated Vortex Methods for the 2D Euler Equations'. In: *Journal of Computational Physics* 111.2 (Apr. 1994), pp. 291–323. ISSN: 0021–9991. DOI: 10.1006/jcph.1994.1065.

[81] G.-H. Cottet and P. Poncet. 'Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods'. In: *Journal of Computational Physics* 193.1 (Jan. 2004), pp. 136–158. ISSN: 0021–9991. DOI: 10.1016/j.jcp.2003.08.025.

[82] Y. Marichal, P. Chatelain, and G. Winckelmans. 'Immersed interface interpolation schemes for particle–mesh methods'. In: *Journal of Computational Physics* 326 (Dec. 2016), pp. 947–972. ISSN: 0021–9991. DOI: 10.1016/j.jcp.2016.09.027.

[83] J. M. Melenk and I. Babuška. 'The partition of unity finite element method: Basic theory and applications'. In: *Computer Methods in Applied Mechanics and Engineering* 139.1–4 (Dec. 1996), pp. 289–314. ISSN: 0045–7825. DOI: 10.1016/S0045-7825(96)01087-0.

[84] C. A. Duarte, D.-J. Kim, and D. M. Quaresma. 'Arbitrarily smooth generalized finite element approximations'. In: *Computer Methods in Applied Mechanics and Engineering* 196.1–3 (Dec. 2006), pp. 33–56. ISSN: 0045–7825. DOI: 10.1016/j.cma.2005.12.016.

[85] E. Burman. 'La pénalisation fantôme'. In: *Comptes Rendus Mathématique* 348.21–22 (Nov. 2010), pp. 1217–1220. ISSN: 1631–073X. DOI: 10.1016/j.crma.2010.10.006.

[86] E. Burman and P. Hansbo. 'Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method'. In: *Applied Numerical Mathematics* 62.4 (Apr. 2012), pp. 328–341. ISSN: 0168–9274. DOI: 10.1016/j.apnum.2011.01.008.

[87] P. Hansbo, M. G. Larson, and S. Zahedi. 'A cut finite element method for a Stokes interface problem'. In: *Applied Numerical Mathematics* 85 (Nov. 2014), pp. 90–114. ISSN: 0168–9274. DOI: 10.1016/j.apnum.2014.06.009.

[88] M. Kirchhart, S. Groß, and A. Reusken. 'Analysis of an XFEM Discretization for Stokes Interface Problems'. In: *SIAM Journal on Scientific Computing* 38.2 (2016), A1019–A1043. ISSN: 1064–8275. DOI: 10.1137/15M1011779.

[89] S. Groß, T. Ludescher, M. Olshanskii, and A. Reusken. 'Robust Preconditioning for XFEM Applied to Time-Dependent Stokes Problems'. In: *SIAM Journal on Scientific Computing* 38.6 (Nov. 2016), A3492–A3514. ISSN: 1095–7197. DOI: 10.1137/15M1024007.

[90] L. Cattaneo et al. 'Stabilized extended finite elements for the approximation of saddle point problems with unfitted interfaces'. In: *Calcolo* 52.2 (June 2015), pp. 123–152. ISSN: 0008–0624. DOI: 10.1007/s10092-014-0109-9.

[91] E. Burman and M. Á. Fernández. 'An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes'. In: *Computer Methods in Applied Mechanics and Engineering* 279 (Sept. 2014), pp. 497–514. ISSN: 0045–7825. DOI: 10.1016/j.cma.2014.07.007.

[92] F. Brezzi and J. Pitkäranta. 'On the Stabilization of Finite Element Approximations of the Stokes Equations'. In: *Efficient Solutions of Elliptic Systems*. Ed. by W. Hackbusch. Vol. 10. Notes on Numerical Fluid Mechanics. Vieweg+Teubner Verlag, 1984, pp. 11–19. ISBN: 9783663141693.

[93] A. Reusken. 'Analysis of an extended pressure finite element space for two-phase incompressible flows'. In: *Computing and Visualization in Science* 11.4–6 (Sept. 2008), pp. 293–305. ISSN: 1432–9360. DOI: 10.1007/s00791-008-0099-8.

[94] J.-C. Suh. 'The evaluation of the Biot–Savart integral'. In: *Journal of Engineering Mathematics* 37.4 (May 2000), pp. 375–395. ISSN: 0022–0833. DOI: 10.1023/A:1004666000020.

[95] W. Li and M. Vezza. 'A hybrid vortex method for the simulation of three-dimensional flows'. In: *Numerical Methods in Fluids* 57.1 (May 2008), pp. 31–45. ISSN: 1097–0363. DOI: 10.1002/fld.1610.

[96] A. van Oosterom. 'Closed-form analytical expressions for the potential fields generated by triangular monolayers with linearly distributed source strength'. In: *Medical & Biological Engineering & Computing* 50.1 (Jan. 2012), pp. 1–9. ISSN: 0140–0118. DOI: 10.1007/s11517-011-0837-9.

[97] *IEEE 754-2008. Standard for Floating-Point Arithmetic*. Institute of Electrical and Electronics Engineers, Aug. 2008. DOI: 10.1109/IEEESTD.2008.4610935.

[98] J. A. Nitsche. 'Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind'. In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36.1 (July 1971), pp. 9–15. ISSN: 1865–8784.

[99] J. A. Strain. '2D Vortex Methods and Singular Quadrature Rules'. In: *Journal of Computational Physics* 124.1 (Mar. 1996), pp. 131–145. ISSN: 0021–9991. DOI: 10.1006/jcph.1996.0049.

[100] P. Ploumhans et al. 'Vortex Methods for Three-Dimensional Bluff Body Flows: Application to the Sphere at Re=300, 500 and 1000'. In: *Journal of Computational Phsyics* 178.2 (May 2002), pp. 427–463. ISSN: 0021–9991. DOI: 10.1006/jcph.2002.7035.

[101] A. Gharakhani and M. J. Stock. '3-D Vortex Simulation of Flow Over A Circular Disk at An Angle of Attack'. In: *Proceedings of the 17th AIAA Computational Fluid Dynamics Conference*. Toronto, Ontario Canada. June 2005.

[102] N. G. Meyers and J. Serrin. '$H = W$'. In: *Proceedings of the National Acadamy of Sciences* 51.6 (June 1964), pp. 1055–1056. ISSN: 0027–8424.