

This page intentionally left blank

“You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete.”

Buckminster Fuller

Acknowledgements

I would like to thank first and foremost my little brother Jayanthan for reading and correcting this cookbook. Coming from the realms of Philosophy it can be quite difficult to read through higher level mathematical descriptions. Also I want to thank Ken Thomson and Dennis Ritchie, who passed away 2011. Both of them were very inspiring to me and led me to pursuit my passion in computer science.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	ix
Abbreviations	x
Physical Constants	xi
Symbols	xii
1 Introduction	1
1.1 The Field of Computational Fluid Dynamics	1
1.2 Typical applications	2
1.3 Focus of this Report	3
1.4 Content Outline and report structure	3
1.5 Conventions	4
1.6 Further reading	4
2 Physical phenomena of fluid dynamics	5
2.1 Physical introduction	5
2.2 Fluid dynamics terminology	5
2.3 Internal Flows	6
2.4 External Flows	6
2.5 Unsteady flow	6
2.6 Incompressible fluids	6
2.7 Inviscid flow	6
2.8 Laminar flow	6
2.9 Turbulent flow	7
2.9.1 Reynolds number	8

3 Numerical Methods	10
3.1 Introduction	10
3.2 Numerical simulation proceedings	10
3.3 Eulerian and Lagrangian perspective	11
3.3.1 Lagrangian Description	13
3.3.2 Eulerian Description	14
3.3.3 Lagrangian vs. Eulerian Description	15
3.3.4 Implications on Fluid Dynamics	16
3.4 Differential equations	16
3.4.1 Ordinary Differential equations	17
3.4.2 Partial Differential equations	17
3.4.3 Divergence theorem	17
3.5 Implicit Methods	17
3.6 Explicit Methods	17
3.7 Boundary Conditions	17
3.7.1 Von Neumann Conditions	17
3.7.2 Dirichlet Conditions	17
3.8 Governing equations of Fluid Dynamics	17
3.8.1 Conservation of Momentum	18
3.8.2 Conservation of Mass	18
3.8.3 Conservation of Energy	18
3.8.4 Euler equations	18
3.8.5 Navier-Stokes equations	18
3.9 The Concept of Discretization	19
3.9.1 Discretization in time	19
3.9.2 Discretization in space	20
3.10 Meshes	20
3.10.1 Structured and unstructured meshes	20
3.10.2 Eulerian Mesh	20
3.10.3 Lagrangian Mesh	22
3.11 Further Reading	23
3.12 Chapter passage	23
4 Meshbased Methods	24
4.1 Meshbased discretization	24
4.2 FDM	24
4.2.1 Basic Idea of FDM	24
4.2.2 Taylor series expansion and error analysis	25
4.2.3 Structured Meshes in FDM	27
4.2.4 Problems of FDM	27
4.2.4.1 Problems with conservation	27
4.2.4.2 Problems with irregular meshes	27
4.3 FVM	27
4.3.1 Basic Idea of FVM	27
4.3.1.1 Conservative property	29
4.3.2 FVM vs. FDM	29
4.4 FEM	30

4.4.1	Advantages of FEM	30
4.4.2	Problems of FEM	30
4.4.2.1	Remeshing	31
4.4.2.2	Problems with hyperbolic equations	31
4.5	Summary of Meshbased Methods	31
5	Smoothed Particle Hydrodynamics	32
5.1	Mesh-free methods	32
5.1.1	Why use mesh free methods?	32
5.1.2	History of SPH	33
5.1.3	Applications of SPH	33
5.1.4	Basic idea of SPH	33
5.2	Mathematical foundation	34
5.2.1	Kernel approximation	35
5.2.2	Particle approximation	37
5.2.3	Lagrangian Navier-Stokes equations	40
5.2.4	SPH vs. FEM	40
5.2.5	Benefits of Meshfree methods over Meshbased methods	40
5.3	Implementation	40
5.3.1	Nearest neighbour particles	40
5.3.2	Issues of computer implementation	40
5.3.3	Parcallel Coding	40
5.4	Implementation of Navier-Stokes equations	40
5.5	Particle consistency problem	40
6	Usage in Simulation Software	41
6.1	Mesh-free methods	41
6.1.1	Why use mesh free methods?	41
6.2	Implementation	41
Bibliography		42

List of Figures

2.1	External flow on around airfoil	6
2.2	Laminar flow	7
2.3	Velocity field of the laminar flow	7
2.4	Turbulent flow	8
2.5	Velocity field of the turbulent flow	8
3.1	A CFD simulation process	12
3.2	Illustration of a glass bottle with Olive oil and spice particles	13
3.3	Schematic diagram of particle's speed and pathlines in Lagrangian perspective	14
3.4	Illustration of hexagonal observer nodes inside a rectangular mesh in Eulerian perspective	15
3.5	Perspective vs. Discretization Methods	21
3.6	bla	22
3.7	bla	23
4.1	Update of Mesh Nodes in FEM	31
5.1	Image sequence of the Formation of Stars and Brown Dwarfs in a Star Cluster	34
5.2	Illustration of particle approximation with kernel function K and quadratic support domain	38

List of Tables

2.1	Table of critical Reynolds numbers	9
4.1	Comparison of FVM and FDM	29

Abbreviations

CFD	Computational Fluid Dynamics
CSM	Computational Solid Mechanics
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
LHS	Left Hand Side
MPM	Meshfree Particle Method
NSE	Navier - Stokes Equations
RHS	Right Hand Side
SPH	Smoothed Particle Hydrodynamics
TSE	Taylor Series Expansion
VFX	Visual Effects

Physical Constants

Speed of Light c = $2.997\ 924\ 58 \times 10^8$ ms⁻¹ (exact)

Symbols

<i>sign</i>	Name	SI Unit
a	distance	meter m
m	mass	kilogram kg
P	power	W (Js^{-1})
t	time	seconds s

ν	kinematic viscosity	$\frac{m^2}{s}$
ρ	density	$\text{kg (m}^{-3}\text{)}$
ω	angular frequency	rads^{-1}

Dedicated to my dear friend Tina. . .

Chapter 1

Introduction

1.1 The Field of Computational Fluid Dynamics

The field of Computational Fluid Dynamics (CFD) is widely utilized in science, engineering and even in visual effects for feature film. This written report gives a introduction to the topic of CFD. First of all to define more clearly what the term CFD means, we start of with the quote:

"CFD (computational fluid dynamics) is a set of numerical methods applied to obtain approximate solutions of problems of fluid dynamics and heat transfer." [Zikanov, 2010, p. 1].

A comprehensive quote that includes the main subjects of CFD: **Numerical methods**, **approximation**, **practical problems**. In the first place, as mentioned, "*CFD (...) is a set of numerical methods (...)*". In other words CFD is based in the realms of mathematics and numerics. To understand the behaviour of fluids and predict it in known circumstances the reader needs to have a strong mathematical background and furthermore be able to transfer this knowledge into practical computation. Second "*(...) obtain approximate solutions (...)*" refers to the nature of problem modelling and solution. Given a fluid dynamic problem (e.g. flow of water in a pipe), there will be always a level of abstraction involved in transferring the real world problem into a computable model. Above that many more levels of approximations are involved, which are due to current limits of science and the discrete nature of computers. Finally "*(...) problems of fluid dynamics and heat transfer.*" outlines the very purpose of CFD, solving real world physical fluid problems. As we will see in the following Section, the wide applications have in common that a real fluid problem needs to be solved and this

often isn't feasible with practical model making or other techniques. Therefore CFD can be understood to be a modern *tool* to tackle fluid problems with the help of computers. This three aspects of CFD will be discussed in this report, albeit in a different order and with a focus on the topic of discretization. In the following Sections of this Chapter the typical applications of CFD are briefly presented and afterwards the focus of this report is discussed in more detail before presenting the report's content outline.

1.2 Typical applications

The demand for analyzing and/or predicting fluid flows is increasing in the fields of engineering and science. Following inchoate list includes some evident industrial and science application examples and also some that might not come into mind when thinking about CFD. For further reading to each field some references are given.

- Aeroacoustics : Analysis of turbulent airflow around automobiles to reduce the noise generation and increase passenger comfort [[Naugolnych and Ostrovsky, 1998](#)].
- Aerodynamics: Optimization of the plane's airfoil design to reduce overall fuel consumption [[Anderson, 2010](#)].
- Atmospheric sciences: Weather prediction (shortterm) [[Warner, 2010](#)] and climate research (longterm) [[McGuffie and Henderson-Sellers, 2005](#)].
- Building Engineering: Air flow simulations with models of planned buildings to optimize heat transfer [[Underwood and Yik, 2008](#)].
- Chemistry: Combustion analysis for internal fuel engines (e.g. car engine) to determine the best ignition frequency for a certain octane rating [[Shi et al., 2011](#)].
- Civil Engineering: Fluid simulations to determine the consequences of dam breaks [[Bates et al., 2005](#)].
- Hemodynamics: Flow analysis of blood in the human heart to determine better ways to treat high blood pressure [[Guccione et al., 2009](#)].
- Magnetohydrodynamics: Simulation of plasma in models of new thermonuclear fusion reactors [[Goedbloed et al., 2010](#)].
- Mechanical Engineering: Flow analysis to optimize the design of turbomachinery (e.g. turbines) [[Elder et al., 2003](#)].

1.3 Focus of this Report

This report gives a overview on the topic of CFD. But it will focus on the mesh-based and mesh-free methods and will in great detail describe latter. Both of these methods are discretization methods. Revisiting the introduction, they can be allocated under the terms numerical methods and approximation. This report emphasis the mathematics and numerics of these methods. Furthermore it will pick out the Smoothed Particle Hydrodynamics (SPH) as a representative of mesh-free methods and will compare it to the mesh-based Finite Elements Method (FEM). Throughout the entire report a unsteady, time-dependend and evolving fluid is assumed.

Two questions should be answered concerning the mesh-based and mesh-free methods:

- item 1) What is the best way to compute strong fluid deformations efficiently?
- item 2)

Since CFD is a huge topic the introduction and physical description chapters of this report will only explain the terms that are relevant to understand the last chapters.

1.4 Content Outline and report structure

This free book provides a comprehensive introduction and overview of computational fluid dynamics. While it focuses on the modern mesh-free SPH approach, the classical mesh based methods like FEM are also explained. The premier goal is to give a novice reader a basic understanding of CFD. The first question that needs to be answered in order to structure this book is: *What are the vital parts of CFD?* Considering this book was first written for computer scientist who often need to understand a new subject in order to implement it, a more approach based question could be easier to answer: *What are the vital parts of CFD that a C.s undergrad needs to understand in order to implement a CFD system?*

To answer this question this book takes a look at the main steps of a general CFD problem solving approach. Since the focus of this report is on discretization methods, the vital parts for undergrad to understand lie not so much in physics but in the fundamental mathematical ideas. Therefore this book introduction briefly dives into the physical phenomena of fluid dynamics. The presented introduction from Chapter 2 through Chapter 3 also emphasises the mathematics more than the physics.

The text is organized in five Chapters that are outlined as follows:

Chapters

- Chapter 1: Introduction and focus of this report
- Chapter 2: The physical world of fluids including important terms
- Chapter 3: Numerical methods that include the most important mathematical laws around fluid dynamics
- Chapter 4: The dominant mesh-based methods
- Chapter 5: Mesh-free methods

1.5 Conventions

Throughout this report the Harvard referencing style is used for quotes and general literature references.

1.6 Further reading

Along this report in each Chapter several references are given to a specific topic. For a comprehensive start with German literature [[Laurien and Oertel, 2009](#)] is a good choice.

Chapter 2

Physical phenomena of fluid dynamics

2.1 Physical introduction

In this chapter the physical basics of fluid dynamics shall be introduced and explained. Since the focus of this Chapter is to give the novice reader a good understanding of the underlying physical and mathematical terms of CFD, at first specific physical phenomenon of fluids are presented and illustrated in detail before slowly moving to the realms of mathematics in this Chapter and the following to further describe these phenomena. We start with the explanation of several important terms of the field and then continue with the governing equations of fluid dynamics by explaining them with the simpler Euler equations in the next Chapter. Afterwards the Navier-Stokes equations are presented as an extension of the Euler equations. In Chapter 3 the given equations are explained in more detail from the mathematical and numerical view.

2.2 Fluid dynamics terminology

First of all the most important terms in fluid dynamics are presented and illustrated. However the field of CFD is very large and not all terms can be covered. This Chapter focuses on the terms that are necessary to understand the following Chapters. For further reading [[Hughes and Brighton, 1999](#)] [[Potter, 2009](#)] and [[Biringen and Chow, 2011](#)] are recommended.

2.3 Internal Flows

2.4 External Flows

A external flow is considered to be a fluid flow that takes places around or outside of a certain geometry body. Examples of such flow occur around an airfoil of a airplane, as illustrated in Figure 2.1 Fluid Mechanics Dem Page 67 Figure 3.6 modified

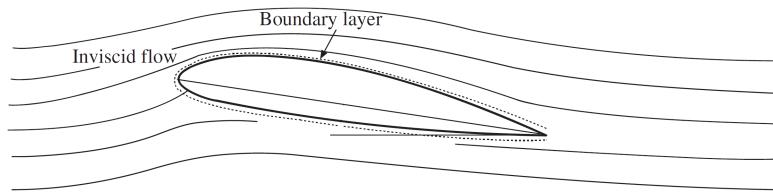


FIGURE 2.1: External flow on around airfoil

For further reading on internal and external flows please refer to [Hirsch \[2007\]](#).

2.5 Unsteady flow

2.6 Incompressible fluids

2.7 Inviscid flow

The ideal flow also known as

2.8 Laminar flow

If viscosity can be considered important in the modelling of the fluid flow, A laminar flow is one in which no or relatively negligible amounts of fluid particles mix together in the course of the flow direction. A illustration of this circumstance can be viewed in Figure 2.2. The coloured streams represent the flow of one fluid along a short pipe. The motion of the coloured streams remains smooth and uniform without random mixing of colours. The fluid travels smoothly along regular paths and the different coloured fluid layers slide over each other without interception.



FIGURE 2.2: Laminar flow

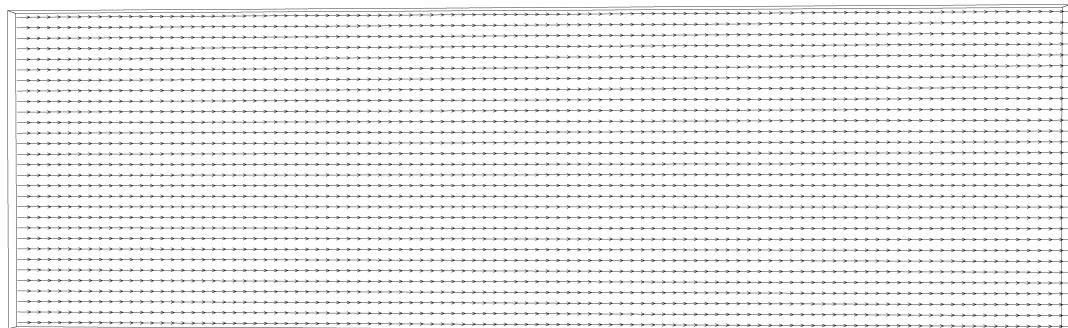


FIGURE 2.3: Velocity field of the laminar flow

2.9 Turbulent flow

Turbulent flow can be considered the opposite of laminar flow. Heavy mixing of fluid particles takes place in a turbulent flow. Important to note is that turbulence is a property of the the fluid flow not of the fluid itself. The motion of the flow itself and each individual particle can be very noisy and random. In Figure 2.4 after a short period of uniform laminar flow the colour streams break up and become turbulent. There is heavy mixing of colour noticeable and the fluid's path become chaotic. This is due to the viscosity of the fluid itself and also the lenght itself of geometry the fluid flows along. In general any kind of random variations in a flow's quantities is considered to be turbulent. In Figure 2.5 this is the velocity: The random fluctuation of the velocity can be seen at a certain moment. All this factors are represented with the Reynolds number

Re which makes it possible to determine if a flow is likely to be laminar or turbulent. Important ist to understand the difference between fluids and flow since

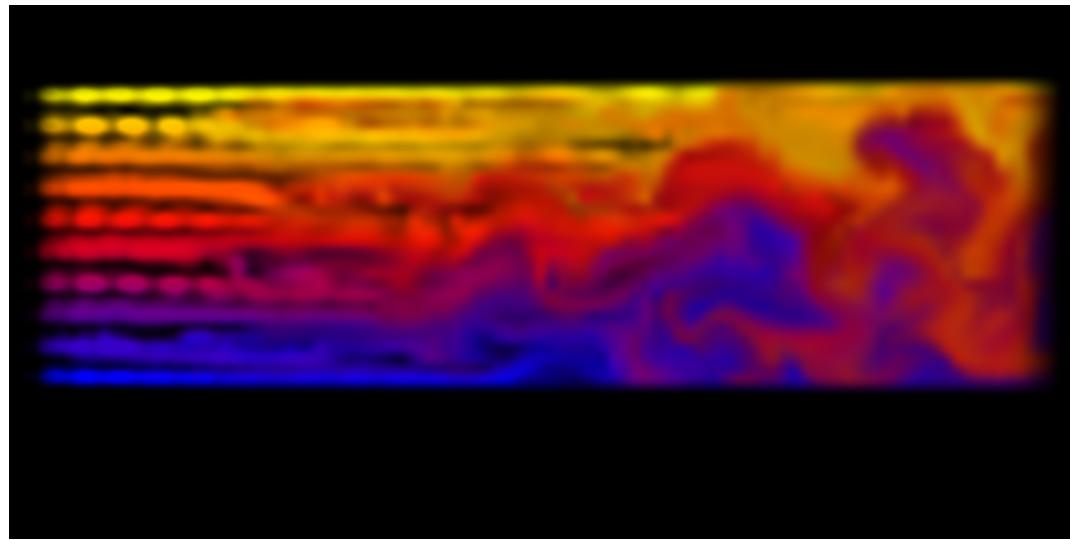


FIGURE 2.4: Turbulent flow

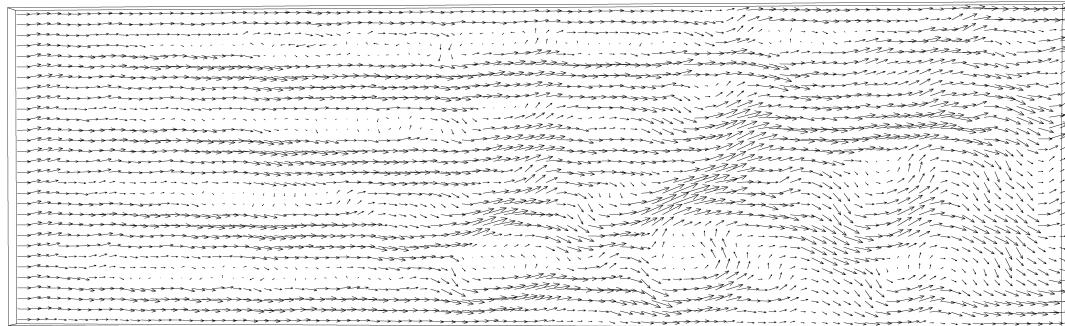


FIGURE 2.5: Velocity field of the turbulent flow

2.9.1 Reynolds number

The dimensionless Reynolds number is used to relate inertial forces to viscous forces in a fluid flow. It can be determined as follows and its value helps to estimate if a flow is more likely to be laminar or turbulent.

$$Re = \frac{VL}{\nu}$$

With V denoting the average velocity of the fluid inside or outside the geometry, L as the characteristic linear dimension or length of the fluid flow (e.g. travelled distance of a fluid) and ν as the kinematic viscosity of the fluid.

The linear dimension L varies greatly between different geometry types and is subject to engineering convention. For instance not the lenght but the diameter of a pipe is used when computing Reynolds numbers for the internal flow in respective geometry today. Therfore when comparing Reynolds number as follows one always needs to consider the geometry for which it was computed and can not easily compare Reynolds numbers of different geometry classes.

Laminar flow occurs in general when the viscous forces are dominant, hence the Reynolds number needs to be low. Turbulent flows occur when the inertial forces are dominant. As rule of thumb if the Reynolds number is relatively large the flow is likely to be turbulent ($Re \gtrsim 4000$), accordingly small numbers are likely to imply a laminar flow ($Re \lesssim 2100$). The term "relative" means that the seperation between large and small Reynolds numbers is dependend on the *critical* Reynolds number. This critical Reynolds number determines the thresholds and is geometry dependend. In

TABLE 2.1: Table of critical Reynolds numbers

Chapter 3

Numerical Methods

3.1 Introduction

In this chapter a overall mathematical introduction to the field of computational fluid dynamics is given. There are several possibilities to classify and structure the wide field of CFD. In most literature this is done by focusing on the physical properties of the fluid models itself e.g. [Date, 2005] resulting in categories like incompressible or compressible fluid dynamics. Since the focus of this report is not on the physical description of fluids, we will take a different approach and structure the report by the domain discretizations schemes itself which result in the later Chapters 3 "Mesh-based Methods" and Chapter 4 "Mesh-free Methods". First of all a brief distinction is made between mesh-based and mesh-free methods. These distinction is followed through the entire chapter and will conclude in detailed descriptions of both methods given in Chapter 3 and Chapter 4. Afterwards the governing physical Navier-Stokes equations of CFD are given and explained in detail. For further reading please refer to [Date, 2005], [Cebeci, 2005], [Toro, 2009] and for a comprehensive overview of the mathematical notations in CFD please refer to [Avrs, 1989].

3.2 Numerical simulation proceedings

Before diving into the physical world of CFD we first take a look at the general approach to solving a physical problem with a numerical simulation. By understanding this process we can lay out the Chapter in a more structured way. Figure 3.1 shows this general process for a fluid flow problem and begins with the physical problem in the real world. Before even considering any kind of computer program or mathematical equation, the physical phenomena that are involved or might appear in the real world have to be

understood. Some important examples for this like *turbulent flow* were given in Chapter 2. From this point on, one needs to abstract from nature in order to define a computable model. Conscious decisions have to be made whether the apprehended phenomena are relevant enough to be involved in the mathematical modelling or not. At the beginning of the mathematical modelling process this Chapter will describe in Section 3.3 two major and different **perspectives** in how a observer can perceive a fluid flow. This **Eulerian** or **Lagrangian** perspective leads to different mathematical equations which will be discussed later. They have vast implications on the discretization process and simulation speed as it will be shown later. After that the best approximation to the abstracted flow problem, the physical laws that influence the narrowed down phenomena need to be identified and mathematically modeled in the appropriate perspective. Usually this leads to mathematical equations, which are in differential or integral form and describe a continuous behaviour of a physical property. In Section 3.4 they are discussed. The most relevant physical laws and equations of fluid flow are given in Section 3.8. In order to be computed by a machine with a finite and discrete number space in finite time, the *space*, *equations* and even *time* (for unsteady flow) need to be discretized. The discretization process will be explained in Section 3.9 and in the following Chapters in great detail, since this is the focus of this report. Afterwards the discretized model needs to undergo a in-depth numerical analysis on the amount and type of errors the discretization has caused in comparision to the continuous model. This will be done in more detail in the specific discretization methods in the following Chapters (e.g. 4.2.2). The last following four parts are not the focus of this report and would go beyond the constraints of it. Therefore introduction literature into these topics is given. The actual **Implementation** would take place, leading to a program that would take care of numerical **Solving** the equations and returning the relevant physical quantities dependent on space and time - the **Simulation**. These results need to be visualized and in many steps analyzed [Oberkampf and Roy, 2010].

Furthermore a comprehensive list of commercial CFD simulation programs is given in [Martin, 2011, pg.139]. To better understand the efficient solving of large sparse matrices please refer to [Saad, 2003].

3.3 Eulerian and Lagrangian perspective

While formulating the governing equations of a physical problem like fluid dynamics, there are two fundamentally different perspectives that can be taken by a observer to describe the behaviour of that physical world. This perspective is also called *observational frame of reference*. The Eulerian perspective depends on a static frame of reference.

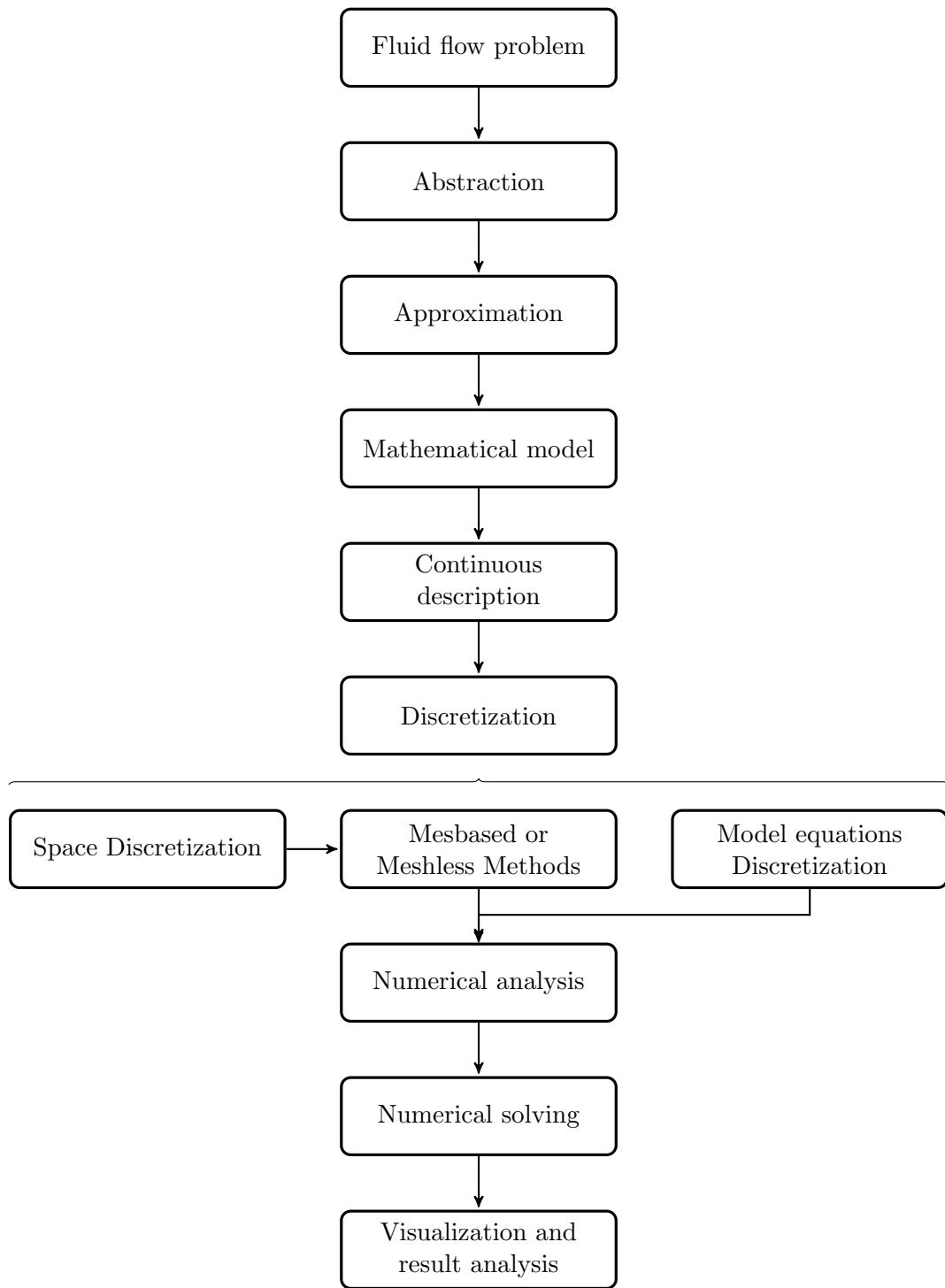


FIGURE 3.1: A CFD simulation process

The Lagrangian perspective depends on a floating frame of reference. From these two perspectives different descriptions of the same problem will result . These have wide implications on the later solving of the formulated equations. They are essential to understand the differences within the Meshbased methods and moreover between them and the Meshfree methods. In this report the term *frame of reference* is not equivalent with the *coordinate system* of a description. Also to shorten up the Eulerian description is often called simply *Eulerian* and accordingly the Lagrangian description is called *Lagrangian*.

To better understand both perspectives, please consider Figure 3.2. It illustrates a glass bottle of olive oil (grey), which is mixed with spice particles like pepper (black)¹. The oil is incompressible and the particles are capable of moving freely around in the glass bottle. The observer is able to look inside the transparent glass bottle and identify single particles. These particles can also be seen as abstraction's of fluid parcel therefore both terms are used in exchange in this Section. The entire bottle is shaken, to give the particles and the entire fluid some initial movement. From time t_0 on the bottle's content is observed, while resting on a table.

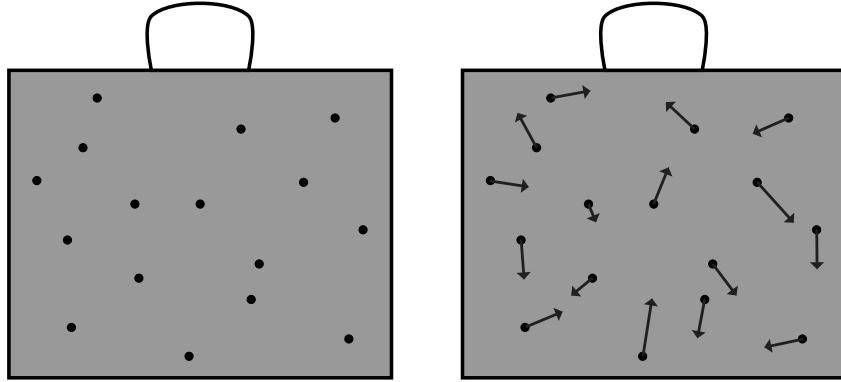


FIGURE 3.2: Illustration of a glass bottle with Olive oil and spice particles

3.3.1 Lagrangian Description

In the Lagrangian perspective the observer focuses on each separate particle or fluid parcel as it moves inside the glass bottle over time. Therefore the Lagrangian is a *material* description. Each particle is *labelled* p_i with $i \in \mathbb{N}_0$ at the time t_0 (resting bottle). The observer moves with each parcel and measures the physical values like velocity (arrow) or mass of it. These values are recorded for each particle p_i . One can

¹Illustration inspired by Luis Cafarelli's lecture on the NSE. Video link to lecture active on 11th of March: <http://vimeo.com/18185364>

consider these particles to be small moving *sensors* that record physical properties of themselves.

When the recorded information is presented of a property like speed, this might schematically look like the LHS of Figure 3.3. Each individual particle has the information of speed at a given time. For the observer this already implies that it is possible to know the fate of each particle and its values for the entire observed time. Furthermore one is able to plot the transport pathways, called *pathlines*, of each individual particle in a transport process as depicted on the RHS of Figure 3.3.

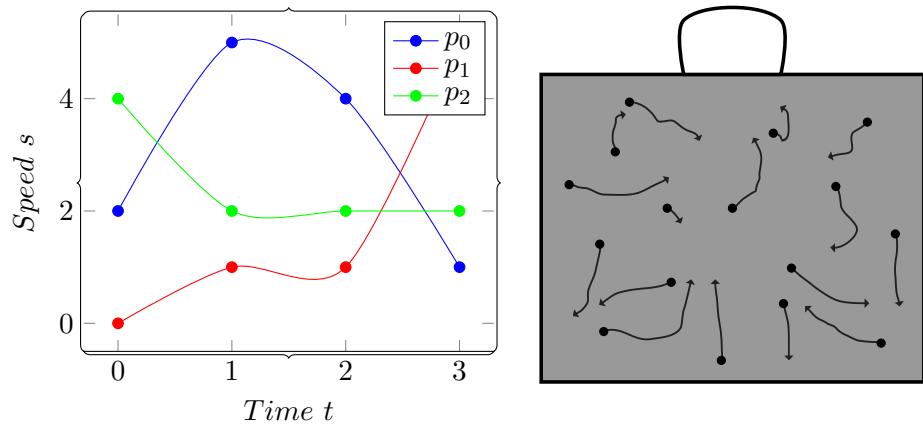


FIGURE 3.3: Schematic diagram of particle's speed and pathlines in Lagrangian perspective

3.3.2 Eulerian Description

In the Eulerian description the observer does not follow individual fluid parcels. He rather rests and focuses on certain selected locations in the domain of the glass bottle. Therefore the Eulerian is a *spatial* description. To better identify these observation locations, a rectangular mesh is drawn onto the bottle with the intersections being the points of interest. In Figure 3.4 this is illustrated with hexagonal observer nodes that are located on the intersection points of a rectangular mesh. This mesh discretizes the spatial domain of the glass bottle (see [Eulerian Mesh](#)). A fluid parcel is only noticed over time and measured by the observer when it passes through a observer node, this is depicted with the red hexagons. So in contrast to the Lagrangian perspective, the *sensors* are static in location. Consequently in the Eulerian the observer is not capable of knowing the trajectories of all fluid parcels.

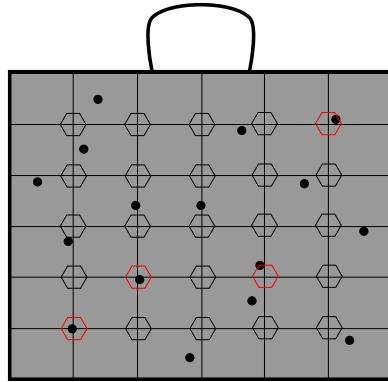


FIGURE 3.4: Illustration of hexagonal observer nodes inside a rectangular mesh in Eulerian perspective

3.3.3 Lagrangian vs. Eulerian Description

To bring both descriptions into a mathematical framework we first start with the Lagrangian. As mentioned above each individual particle is labeled p_i at time t_0 when the bottle is released to rest. This label is a vector of the fluid properties or the particles Cartesian coordinates at the time of release. The Lagrangian observer follows the particle after time t_0 and notes down the position of this individual particle at time t as:

$$X_i(p_j, t_0 | t) \quad (3.1)$$

In Eulerian description the position of a observer is simply x_i . The above situation that this Eulerian observer is able to recognize a fluid parcel (red hexagons) is only given when:

$$x_i = X_i(p_j, t_0 | t) \quad (3.2)$$

Similar to this the velocity relation in Eulerian and Lagrangian perspective is given by:

$$v(x_i, t) = \frac{\partial X_i}{\partial t}(p_j, t_0 | t) \quad (3.3)$$

Where the LHS denotes the velocity v from the fluid parcel detected by a Eulerian observer, the RHS denotes the Lagrangian version. In similar manner the relation to the rate of change of other physical quantities like heat or momentum can be found. A

important mathematical instrument for this is the Stokes derivative, also called material derivative.

3.3.4 Implications on Fluid Dynamics

In Fluid Dynamics the transport or *flux* of fluids is of primary interest. In Eulerian description the flux can only be estimated correctly when there is a dense spatial and temporal coverage of observing nodes.

Flux processes itself are fundamentally a Lagrangian quantity and therefore easier to describe in this perspective. As we will see in the following Chapters, methods with a Eulerian description like FVM have ways to estimate the flux in a certain region. But with more complicated, even deforming, geometry the Lagrangian description is capable of handling these changes far better (e.g FEM). The Lagrangian is used widely in astrophysics, where the gravity of stars, planets and galaxies interact within a vast space of vacuum without any interesting behaviour. The focus lies here on relatively few objects/parcels in comparison to the immense spatial distribution of those. Concerning CFD a rough estimation can be already made that the Lagrangian is better for (many) single fluid parcels that are of interest and can have great deformation behaviour. As we will see in this report this is not entirely true. A method in Lagrangian description and a attached mesh can lead to more problems with deformations in CFD than a Eulerian method (see [FEM](#)). In addition leaving out a mesh in a Lagrangian method, as it is done in [Smoothed Particle Hydrodynamics](#), brings out the deformation robustness concerning CFD. For further reading please refer to [\[Bennett, 2006\]](#).

3.4 Differential equations

In the following Section the [Governing equations of Fluid Dynamics](#) are presented. These equations rely heavily on higher mathematics and describe the fluids properties and flow with various differential equations. This section should brush up very briefly the knowledge on differential equations for the novice reader. Of utmost significance is the classification of partial differential equations into parabolic, hyperbolic and elliptic types, since the later numerical discretization methods are not equally capable of solving all of them efficiently.

3.4.1 Ordinary Differential equations

Ordinary Differential equations (ODE) occur in many CFD terms. Of special interest for this report are they because they are present in the [Smoothed Particle Hydrodynamics](#) method.

3.4.2 Partial Differential equations

3.4.3 Divergence theorem

3.5 Implicit Methods

3.6 Explicit Methods

3.7 Boundary Conditions

3.7.1 Von Neumann Conditions

3.7.2 Dirichlet Conditions

3.8 Governing equations of Fluid Dynamics

As we have seen in Chapter 2 there are many different types of flow in Fluid Dynamics and many physical properties that define them. As this report focuses on unsteady flow, a key finding of Science is that the *entire* motion and behaviour of a fluid flow can be indicated by the conservation laws for **momentum**, **mass** and **energy**. We will use the already explained general form of conservation laws and apply them to these three properties leading to the Euler equations and Navier-Stokes equations. These equations are of a very general Nature and need supporting physical laws when they are applied to a more specific fluid type like [Incompressible fluids](#). This can lead to a massive simplification of the NSE, since depending on the nature of the specified fluid type many equations become simpler or even zero out. Moreover also external properties like forces need be considered, when they actually influence the fluid itself. An excellent example of such can be found in [Warner, 2010] where in order to model the behaviour of fluid packets in our atmosphere also the coriolis forces and Earth's gravity influence the fluids momentum. The set of equations are usually coupled to a certain degree, which complicates solving them.

3.8.1 Conservation of Momentum

3.8.2 Conservation of Mass

The conservation of mass is a key property of fluid flow. In a closed fluid system no mass is created at any point or destroyed. This fact is of great use later on in FVM were withing a subdivided discrete volume no material sources or sinks exists. The physical changing value of the fluid's density ρ is bound by this law in the overall system. The fluids mass only is transported by convective flux, which considering the above notations of a conservation law leads to:

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_A \rho \vec{v} \cdot d\vec{A} = 0 \quad (3.4)$$

3.8.3 Conservation of Energy

Mathematical model of fluid motion Form? integral form ...auch sections machen?

3.8.4 Euler equations

The Euler equations don't cover viscosity and any kind of heat conduction effects. They are of hyperbolic nature in space and time. Therefore these equations only cover [Inviscid flow](#) of fluids.

$$\rho \frac{dv}{dt} = \rho((\nabla \cdot v) \cdot v + \frac{\delta v}{\delta t}) = f - \nabla p \quad (3.5)$$

3.8.5 Navier-Stokes equations

The Navier-Stokes equations (NSE) are a set of equations to describe (in)compressible viscous fluid flows. Therefore they can be seen as a extension and a more accurate version of the Euler equations that now include viscosity and heat conduction. Formulated by the French physicist Claude-Louis Navier with contributions from the Irish physicist George Stokes in the 19th century these set of equations are the cornerstones of today's understanding of fluid dynamics. The set of Navier-Stokes equations are in general parabolic equations in time and space.

3.9 The Concept of Discretization

The governing ordinary and partial differential equations that occur in many domains of physics describe *continuous* models. Solving those equations analytically would result in exact continuous solutions. However in practice an exact analytical solution is not possible and needs to be replaced by an approximation that is found numerically with the aid of computers. Modern digital computers are only capable of representing a finite and discrete amount of numbers with a finite amount of computational time, memory and storage. Therefore any continuous model has to be transferred into a discrete model. This process is called *discretization* and for a general spatial or temporal domain Ω this is called *domain discretization*. When this transfer happens several different kinds of error are introduced into the approximate solution. Among others the most important ones are the **number representation error**, **roundoff error** and **truncation error**. For further reading on these error types please refer to [Press, 2007, pg.8].

This section continues the numerical discussion of the CFD equations by focusing on the discretization process. The *continuous* Navier-Stokes equations need to be discretized in order to solve them. Since this report focuses on unsteady fluids, the given equations are time-dependend. The differential equation in CFD always have a spatial component and in our case also have time dependent components in them. In the process of discretization these dimensions have to be discretized. This results in approaches for **spatial** discretization and **time** discretization. The spatial discretization can be roughly divided into Meshbased and Meshfree methods which are the focus of this report. A very brief outline into time discretization is given to better understand the proceedings of this report. This Chapter concludes with the spatial discretization methods which are continued in the following Chapters.

3.9.1 Discretization in time

The discretization of time in differential equations is necessary when time is present in the equations, and furthermore the physical value κ is changing over time. A subdivision in space, can acquire information from neighbouring divisions in all directions. In contrast to this the discretization in time can only use previously computed, known values from the past (κ_{t-1}, \dots) and the present computed value κ_t to predict future values of it. How this is done is not the scope of this report. Important to mention are two different types of methods: The *explicit* methods and the *implicit* methods. The explicit methods are easier to program and involve only a small numerical effort per time step δt . Unfortunately they are numerically not very stable and for most problems need very small time steps δt . The implicit methods are much harder to program since they

involve the solution of equation systems. This leads to a much higher numerical effort for one time step δt . The benefit of implicit methods is that they are numerically stable and much larger time steps can be chosen. Which method is chosen for a CFD problem depends on the requirements of the solution and e.g. larger time steps are necessary. For further reading on this topic [Laurien and Oertel, 2009, pg.63] provides a excellent introduction.

3.9.2 Discretization in space

The spatial discretization of a domain Ω involves subdividing the continuous spatial distribution of a value (e.g. the distribution of temperature values on a heated iron plate) into finite entities. How this subdivision is done and how their properties like position are defined, depends first of all on what kind of mathematical perspective one takes concerning the frame of reference. This can be either a Eulerian perspective or a Lagrangian perspective as described in 3.3. These finite entities that replace this continuum in space can be furthermore bound by [Meshes](#) (Meshbased methods) or move freely in the form of particles (Meshfree methods). The main difference between meshbased and meshfree methods lie in the question if a overall geometrical description, aka a **mesh**, defines the domain discretization or not. To explain this further please consider Figure 3.5. The Meshbased methods are coloured in red, the Meshfree methods in blue. In general the Meshbased methods, like the Finite Difference and Finite Volume method use a Eulerian perspective and consequently have a static frame of reference. The Finite Element methods are also Meshbased methods, but are a hybrid of the Eulerian perspective and the Lagrangian perspective. A mesh is present, but this mesh actually moves along with the properties in interest. The Smoothed Particle Hydrodynamics method is a pure Lagrangian method with a dynamic frame of reference - the moving particles.

3.10 Meshes

In CFD literature often the term *grid* is also used to refer to a mesh. In this report we will stick to the commonly used term mesh.

3.10.1 Structured and unstructured meshes

3.10.2 Eulerian Mesh

The Eulerian Mesh incorporates the concept of the [Eulerian Description](#) in its mesh description. Its reference system is fixed in space. Hence the mesh does not move with

- - - - - The Eulerian perspective
- The Lagrange perspective

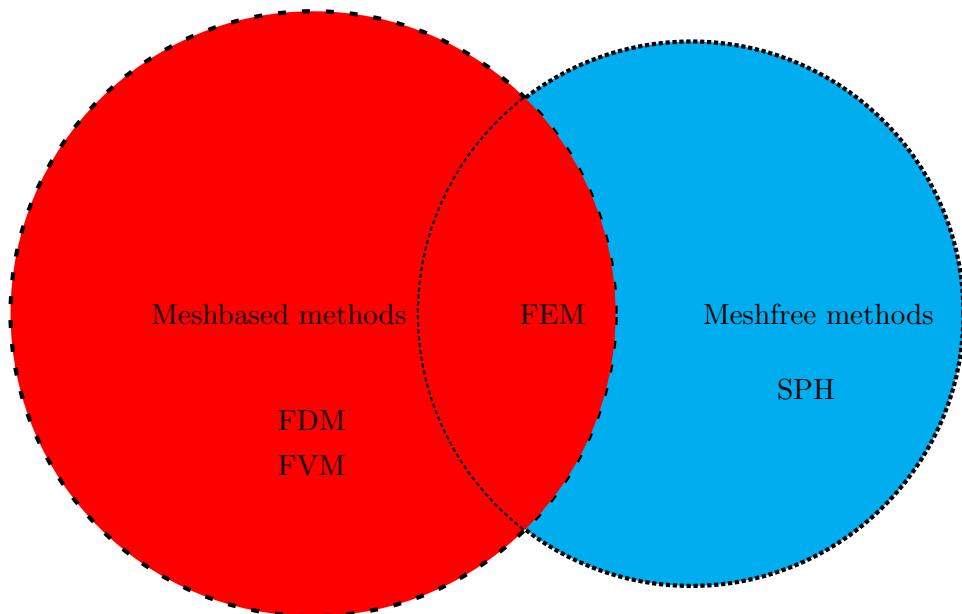


FIGURE 3.5: Perspective vs. Discretization Methods

the material which is simulated, on the contrary the material moves *through* the Eulerian mesh. Depending on the discretization method the mesh is spatially divided into a certain finite amount of subdomains which are called for example *nodes*, *cells* or *volumes* (FVM). These subdomains remain fixed in space and their position, shape and volume usually does not change over the time of the simulation. It is mainly used in strong correlation with the Finite Difference and Finite Volume Methods and its properties will be more discussed in the following Chapters. This mesh type is predominantly used in applications where the deformations of a material are not of great concern and the focus lies for example on the flow behaviour. It is yet important to understand that basically all solvable fluid dynamic problems can be discretized with a Eulerian Mesh, even the ones with strong deformations during simulation. This can be achieved with extremely fine meshes. As it will be shown in Chapter 4.2.4 this imposes skyrocketing requirements on computation time and memory. Figure 3.6 illustrates the concept of the Eulerian mesh. The black, equidistant grid represents the Eulerian mesh, which stays fixed in space throughout a schematic deformation simulation of a material marked in blue. In reality this coarse mesh would not have had the sufficient discretization to adequately capture even this simple deformation. At least a much finer discretization of the deforming edges would have been necessary. Comparing this simple example to the following Lagrangian Mesh already shows that Eulerian Meshes are not very useful for deforming solid mechanics. The analogy of deformations in solid mechanics are

turbulent flows in CFD. Likewise a finer (local) discretization is needed to represent these fine grained physical effects.

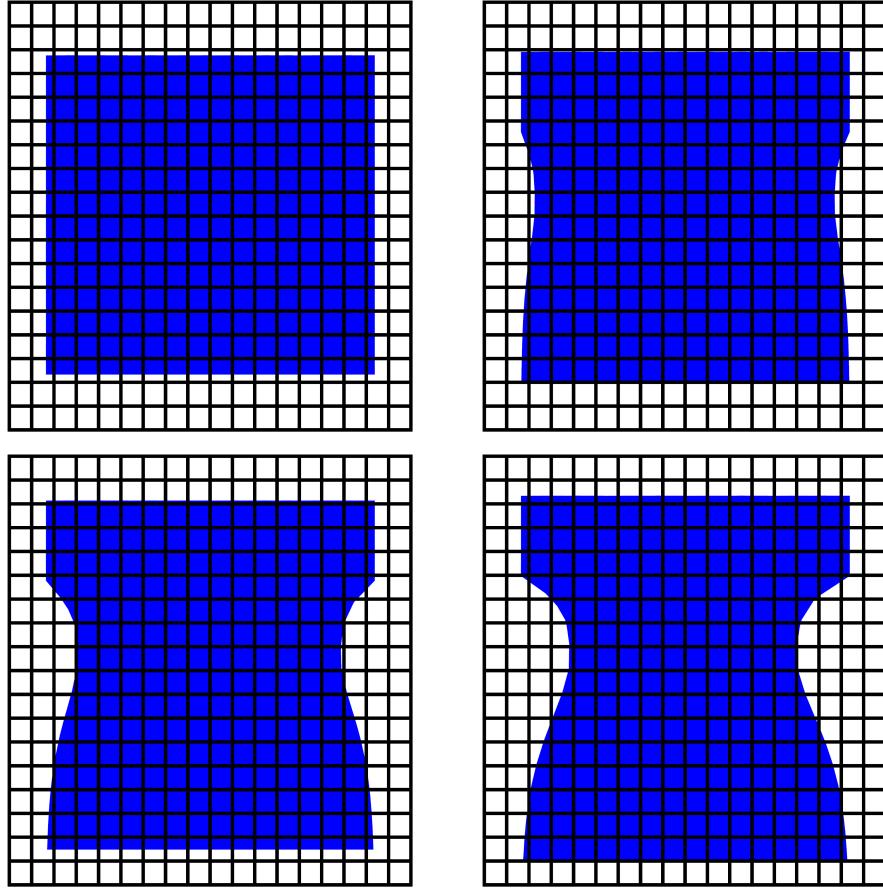


FIGURE 3.6: Illustration of static Eulerian Mesh during material deformation ²

3.10.3 Lagrangian Mesh

The Lagrangian Mesh type is very similar to the Eulerian Mesh except that it incorporates the idea of the [Lagrangian Description](#) into its spatial decomposition: A moving reference system. The mesh itself is clamped to the material and will deform and move accordingly as the material itself. Figure 3.7 demonstrates such a deformation, in which a stretch behaviour is simulated. The mesh is *body-fitted* to the actual geometry that deforms. In contrast to a Eulerian meshes we can clearly see how the triangular elements of the material deform from the left upper picture to the right lower picture during the simulation. This type of mesh is used in the Finite Element Method for CSM and CFD and has various advantages over Eulerian Meshes, which will be discussed in Chapter 4.4.1. However combining a mesh with the Lagrangian perspective also leads to

²Illustration created in Autodesk Maya. This coarse mesh would not be sufficient enough to compute the detail of the deforming blue solid material.

problems like [Remeshing](#). These problems can be overcome by discarding the concept of a mesh description for discretization and creating a meshfree and purer Lagrangian discretization concept with freely moving particles, which is done in Chapter 5. For further reading please refer to [[Bennett, 2006](#)].

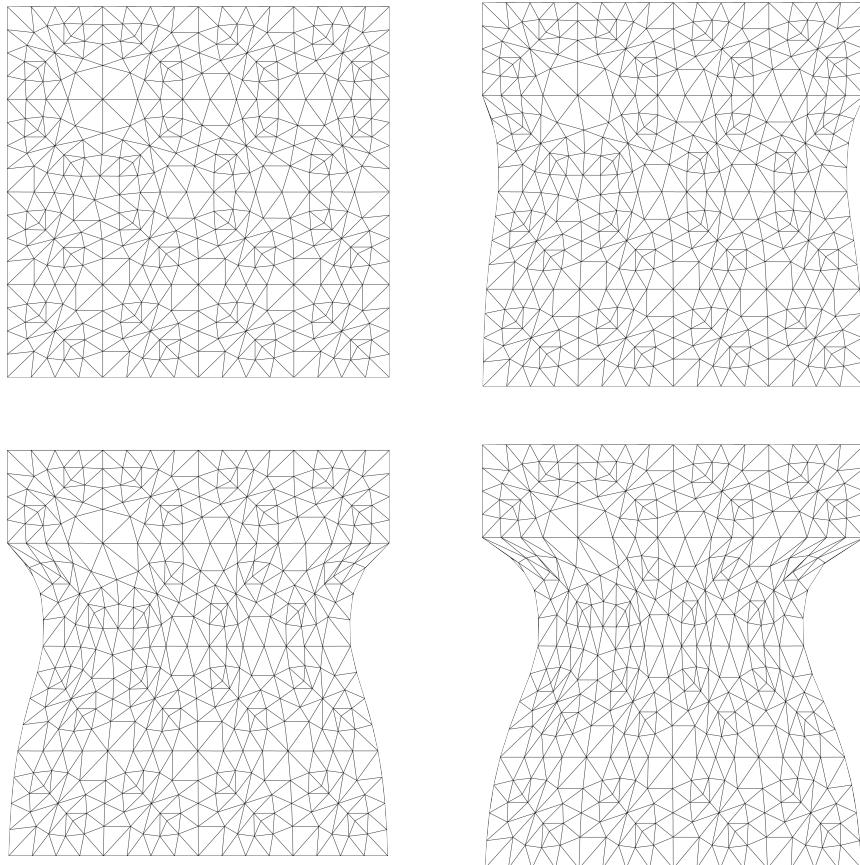


FIGURE 3.7: Movement of Lagrangian Mesh during deformation ³

3.11 Further Reading

Alongside the already given literature in this Chapter, [[Wohlmuth, 2001](#)] and [[Hirsch, 2007](#)] give a in-depth view into discretization.

3.12 Chapter passage

With the now established general knowledge of spatial discretization and mesh types, the following Chapters will describe the actual discretization methods and the mathematical details behind them.

³Illustration created in Autodesk Maya.

Chapter 4

Meshbased Methods

4.1 Meshbased discretization

In this chapter the three most important meshbased discretization methods are introduced and explained in great detail. For further reading on these methods [[Ferziger and Perić, 2002](#)] provides a excellent overview.

4.2 FDM

The Finite Difference Method (FDM) is the easiest and and oldest discretization method. The occurring partial derivatives in the CFD equations are replaced or better approximated by algebraic finite difference quotients. The approximations of the derivatives rest on Taylor series expansions and the method will be explained with a simple example before moving to the NSE equations. For further, extensive reading on FDM [[Pozrikidis, 2009](#), chap. 8] is recommended.

4.2.1 Basic Idea of FDM

Given a function $f(x)$ the derivative ∂f at a certain point x can be defined by

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h} \quad (4.1)$$

Now to bring this function into a finite and computable space, the limit is replaced by a sufficiently small and finite h :

$$\frac{\partial f}{\partial x} \approx \frac{f(x+h) - f(x)}{h} \quad (4.2)$$

The continuous function 4.1 has now been discretized to the approximation 4.2. By multiplying this equation with h on both sides and solving it to $f(x+h)$ we now receive:

$$f(x+h) \approx f(x) + \frac{\partial f}{\partial x} h \quad (4.3)$$

In this form the left hand side (LHS) represents the value of the function itself at a point $(x+h)$ and this value can be approximated by the term on the right hand side (RHS). This simple form 4.3 of discretization is called *Euler's Method*. It is the simplest Taylor Series expansion (TSE) of a function when only the first Taylor polynomial in the expansion is used. Thus FDM is heavily based on TSE and in order to better understand the types and magnitude of errors that occur by discretizing a function with finite differences the following section will discuss the Taylor series analysis of FDM.

4.2.2 Taylor series expansion and error analysis

The exact TSE of the above function $f(x)$ when developing it around x with the step h is:

$$f(x+h) = \sum_{i=0}^{\infty} \frac{\partial^i f}{\partial x^i} \frac{h^i}{i!} = f(x) + \frac{\partial f}{\partial x} h + \frac{\partial^2 f}{\partial x^2} \frac{h^2}{2!} + \frac{\partial^3 f}{\partial x^3} \frac{h^3}{3!} + \dots \quad (4.4)$$

The RHS of the equation 4.4 is again a continuous and exact description of the function itself. Even for very large arbitrary values of h this description would give an exact solution at the point $x+h$. In order to discretize and gain a finite amount of terms on the RHS that can be evaluated by a computer, the infinite series needs to be truncated. The type of error that this truncation introduces is called **truncation error**:

$$f(x+h) = \underbrace{f(x) + \frac{\partial f}{\partial x} h}_{\text{Approximation}} + \overbrace{\frac{\partial^2 f}{\partial x^2} \frac{h^2}{2!} + \frac{\partial^3 f}{\partial x^3} \frac{h^3}{3!} + \dots}^{\text{Truncation error}} \quad (4.5)$$

This notation can be rearranged to better understand the meaning of the Truncation error:

$$\frac{\overbrace{f(x+h) - f(x)}^{\text{finite difference}}}{h} - \underbrace{\frac{\partial f}{\partial x}}_{\text{derivative}} = \overbrace{\frac{\partial^2 f}{\partial x^2} \frac{h}{2!} + \frac{\partial^3 f}{\partial x^3} \frac{h^2}{3!} + \dots}^{\text{Truncation error}} \quad (4.6)$$

The truncation error in equation 4.6 can now be better understood as the difference between the discretized finite difference and the actual continuous derivative. Before moving on the actual grids, one needs to understand the important relationship between the grid spacing h and the truncation error which can be denoted with the Landau notation. As it will be shown in the following part, the truncation error is a strong indicator of how fast the error of a finite difference approximation decreases when the grid spacing h is decreasing.

Rewriting equation 4.6 and solving it for the finite difference shows:

$$\frac{f(x+h) - f(x)}{h} = \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2} \frac{h}{2!} + \frac{\partial^3 f}{\partial x^3} \frac{h^2}{3!} + \dots \quad (4.7)$$

Under the assumption that $h \leq 1$ the truncation error in equation 4.7 can be reduced to its most dominant term which is of linear nature. This leads to the Big - \mathcal{O} notation which denotes that the truncation error is proportional to the the grid spacing h .

$$\frac{f(x+h) - f(x)}{h} \cong \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2} \frac{h}{2!} = \frac{\partial f}{\partial x} + \mathcal{O}(h) \quad (4.8)$$

When focusing on the most LHS and RHS of equation 4.8 the connection between the FDM approximation in equation 4.2 and the error analysis lead to the observation, that this approximation and its error is proportional and therefore of *first order*. At the beginning of this section the equation 4.1 used a one-sided step $x + h$ forward in the general description of the derivative. Therefore this difference formula is called *one-sided forward difference of first order*.

The same technique can be used for a backward difference with the step $x - h$. But this will also only lead to a first order difference. Both of these one sided, single step methods are as already stated know as *Euler's Method* and are usually not used in high efficient engineering CFD implementations. To achieve a faster truncation error approach to zero when decreasing the grid spacing h multiple steps have to be made and combined. A simple example of this is the central difference method that adds up the forward difference and the backward difference and hence has a accumulated step of $2h$. This leads to a second order approximation:

$$\frac{f(x+h) - f(x-h)}{2h} = \frac{\partial f}{\partial x} + \mathcal{O}(h^2) \quad (4.9)$$

In the following section the implications of the first and second order methods concerning the mesh itself are illustrated. Above Euler's Method and the central difference method, there are many other multi step methods that are used today. For further reading please refer to [Milne-Thomson, 2000].

4.2.3 Structured Meshes in FDM

4.2.4 Problems of FDM

4.2.4.1 Problems with conservation

4.2.4.2 Problems with irregular meshes

4.3 FVM

The Finite Volume Method (FVM) is the standard discretization and solving method for CFD. It is based on the concept of conservative discretization, which will be discussed in this chapter. In contrast to the previous discussed FDM which used the differential form of a equation for discretization, the FVM uses the *integral* form of such. And transfers volume descriptions into surface integral descriptions with the divergence theorem. As will be shown in this Chapter this method is far superior and more intuitive for applications like CFD since its approximations inherently satisfy the conservation conditions of equations which leads it to be much more robust when it comes to complex unstructured meshes and deformations. Above that using the integral form makes the usage of Neumann boundary conditions much easier. For further reading please refer to [LeVeque, 2002].

4.3.1 Basic Idea of FVM

The simplest and most striking fact of the NSE is that most of the equations are used to **conserve** some quantity like momentum, mass or energy. To simulate a fluid flow these conservation equations need to be solved. The name FVM is based on the idea that a domain is subdivided into a finite amount of volumes. To start diving into the integral descriptions of FVM, first consider κ to be a arbitrary density of a conservative quantity and Ω a certain control volume in R^3 .

To describe the total amount of κ inside Ω it can be written in integral form as:

$$\int_{\Omega} \kappa d\Omega \quad (4.10)$$

When Ω moves through space, it not only changes its location but might also change its shape dependent on time t . Therefore if we want to denote an overall change of the quantity κ inside of Ω , equation 4.10 needs to be changed to:

$$\frac{d}{dt} \int_{\Omega(t)} \kappa d\Omega \quad (4.11)$$

Within the control volume Ω , a **conservative** quantity κ neither gets destroyed (sink) nor created (source). Hence the total amount of κ within Ω can only change by the *flux* in and out of the volume boundaries of Ω . Let the RHS be a generalized change in flux:

$$\frac{d}{dt} \int_{\Omega(t)} \kappa d\Omega = RHS \quad (4.12)$$

When the shape of Ω changes over time, this results in the boundaries in (x, y, z) being time dependent. By decomposing the LHS of 4.12 with the Leibniz parameter integral rules into the boundaries we get:

$$\begin{aligned} \frac{d}{dt} \int_{\Omega(t)} \kappa d\Omega &= \frac{d}{dt} \int_{x_0(t)}^{x_1(t)} \int_{y_0(t)}^{y_1(t)} \int_{z_0(t)}^{z_1(t)} \kappa d\Omega \\ &= \int_{\Omega(t)} \frac{\partial \kappa}{\partial t} d\Omega + \left[\int_{A_x(t)} \left(\kappa \frac{\partial x}{\partial t} \right) dA_x \right]_{x_0(t)}^{x_1(t)} \\ &\quad + \left[\int_{A_y(t)} \left(\kappa \frac{\partial y}{\partial t} \right) dA_y \right]_{y_0(t)}^{y_1(t)} + \left[\int_{A_z(t)} \left(\kappa \frac{\partial z}{\partial t} \right) dA_z \right]_{z_0(t)}^{z_1(t)} \end{aligned} \quad (4.13)$$

Where A denotes the area of the volume in one dimension. Because all the spatial partial derivatives (x, y, z) are derived by time, they can be considered to be the velocity components u_x, u_y, u_z of the volume Ω itself. Rewriting the LHS with the closed surface integral notation leads to:

$$\frac{d}{dt} \int_{\Omega(t)} \kappa d\Omega = \int_{\Omega(t)} \frac{\partial \kappa}{\partial t} d\Omega + \oint_{A(t)} \kappa \bar{u} d\bar{A} \quad (4.14)$$

Thereby \bar{A} denotes the entire closed surface of the volume Ω and the vector \bar{u} denotes the velocity of the volume. This equation is a form of Reynolds transport theorem and of utterly importance to the conservation idea in FVM, for further explanation please refer to [Wesseling, 2009, pg. 10]. Using the divergence theorem equation 4.14 with the RHS from 4.12 the equation can be formulated to a integral form:

$$\frac{d}{dt} \int \kappa d\Omega = \int \left[\frac{\partial \kappa}{\partial t} + \nabla \cdot \kappa \bar{u} \right] d\Omega = RHS \quad (4.15)$$

4.3.1.1 Conservative property

For further reading please refer to [Versteeg and Malalasekera, 2007] and [LeVeque, 2002].

4.3.2 FVM vs. FDM

This comparison gives a brief overview of the differences and similarities between FVM and FDM:

	FVM	FDM
Conservation laws discretization	Integral form	Differential form
Value type	Cell averaged values	Local function values

TABLE 4.1: Comparison of FVM and FDM

FMV has several advantages over FDM, which are summarized in the following list:

- Flexible spatial discretization
- The integral form is more intuitive than the differential form
- FVM is closer to the physics of a fluid flow

A main disadvantage is that FMV does not have a standardized formal theory to analyze the accuracy, like FDM uses with Taylor series expansion and analysis. Still FVM can be "forced" into a structured equidistant mesh which allows a comparative accuracy analysis again FDM.

4.4 FEM

The Finite Element Method (FEM) is usually used in solid and structural mechanics. In contrast to FDM/FVM the Finite Element Method uses also a Lagrangian perspective. Considering Figure 3.5 FEM is located between the Eulerian and Langrangian perspective and is Meshbased. It has indeed properties from both perspectives in his mathematical model. For example in contrast to FDM and FVM, this method uses a **Lagrangian mesh**, meaning that its mesh is attached to the moving material and moves along with it. Similar to FDM/FVM a division of a continuum into finite domains takes place. These subdivisions of the continuum are called finite discrete elements or simply *elements*. These elements are bound by the mentioned topologial Lagrangian mesh which forbids the penetration or overlapping of similar elements. Each element is used to represent a certain part of the material. During the movement of the mesh, the elements stay associated with their location on the mesh and conserve the properties like mass of themselves. For further reading please refer to [[Zienkiewicz et al., 2005](#)], [[Wendt et al., 2009](#), chap. 10]

As mentioned previously extremely high mesh resolution is required to obtain numerical solutions at the smallest turbulence scales. This is very expensive and presently not possible for high Reynolds number flows. It is, therefore, obvious that other alternatives are necessary to obtain a viable approximate solution. The current

4.4.1 Advantages of FEM

4.4.2 Problems of FEM

In situations where the solid or fluid undergoes heavy deformation, the mesh and each of its elements will also become highly deformed and distorted. This is very likely to happen, especially in fluids with lower viscosity. Because of this behaviour certain elements can become numerically *inverted*, afterwards representing a negative mass or volume. To avoid this behaviour during a CFD simulation, the mesh needs often needs to undergo the computationally intensive process of [Remeshing](#). Another way to avoid this is not to use FEM at all and remain with a Eulerian method like FVM.

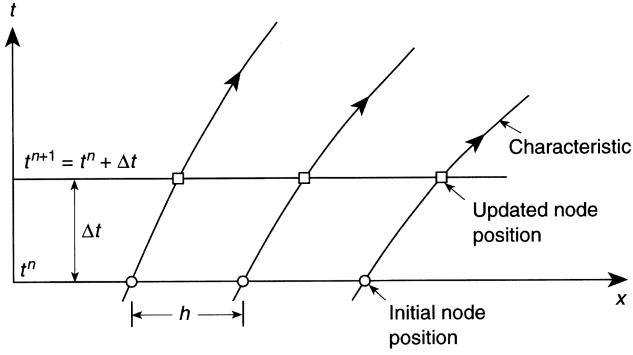


FIGURE 4.1: Update of Mesh Nodes in FEM

4.4.2.1 Remeshing

4.4.2.2 Problems with hyperbolic equations

4.5 Summary of Meshbased Methods

One clear advantage of FDM and FVM over FEM is the simplicity of the Eulerian mesh. Because the mesh stays fixed in space, independent from the fluids flow, these methods are also useful for highly deformations. But they come with a high computational costs, since in order to represent these deformations very fine meshes have to be used. This results also in much higher memory consumption that might lead to a limit what kind of fluid problem can be accurately simulated on given hardware. Furthermore FDM and FVM come with a inherently problem of domain communication. In contrast to FEM, both methods need to transport fluid matter from one mesh subdivision to the next during a flow process. This leads to a always necessary numerical communication between neighbouring subdivisions, no matter if a fluid is present in that cell or not. Consequently the domain decomposition on parallel systems leads to a more complicated communications between cluster machines.

Method	Discretization of	@
FDM	differntial form of conservation equations	@
FVM	integral form of conservation equations	@

Chapter 5

Smoothed Particle Hydrodynamics

5.1 Mesh-free methods

In this chapter the meshfree methods are introduced and explained on the example of the Smoothed Particle Hydrodynamics method. For further reading [[Liu and Liu, 2003](#)] is recommended.

5.1.1 Why use mesh free methods?

As described in Chapter 3 the scientific and engineering community have used mesh-based methods for very long as they have clearly matured over the years. Still Capability to work from micro to macro scale applications.

The benefits of mesh-free methods versus classical mesh-based methods can be outlined as:

- Mesh-free methods can easily handle strong deformations that would lead to remeshing in a FEM simulation. This is due to the absence of a topological mesh which in course of a deformation simulation would lead to problems in connectivity.
ShafonLi 15
- The mesh-free discretization with particles can provide a better representation of a varying geometric shape.

In this report we will focus on the Smoothed Particle Hydrodynamics method as one of the earliest and vastly used mesh-free methods.

5.1.2 History of SPH

The idea of SPH was at first formulated in the year 1977. Its origins do not lie in CFD but in the attempt to model and solve astrophysical problems in a new way. Initially Gingold and Monaghan [[Gingold and Monaghan, 1977](#)] invented and named this approach at the Institute of Astronomy in Cambridge. Their primary intention was to find a approach that would be less computational intensive than FDM to solve various evolution processes (e.g. a star forming from interstellar matter). In their article they state how limited the use of FDM is for their often asymmetrical/distorted problems, since in order to be computed accurately they needed a excessive amount of grid points (resp. high spatial discretization). They proposed a new way of discretization that was based on a Lagrangian description and used a fixed amount of randomly distributed particles. To model physical scalar and vector fields and to retrieve those values during computation they used statistical methods (Monte Carlo integral determination) along with a *kernel function*. Their approach will be described in more detail in the following sections. Inspired by the colleagues at Cambridge, Lucy [[Lucy, 1977](#)] used this numerical approach to test successfully the feasibility of the fission hypothesis for protostars, which is a gas dynamical problem of rotating fluids in space. In contrast to CFD, these problems like the formation of galaxies or solar systems are of macroscopic scale. Today SPH and its further developments are widely adapted in astronomy, CFD and even VFX.

5.1.3 Applications of SPH

Because of several numerical advantages of SPH it is today not only used in astrophysics but also in a wide range of simulations for fluid dynamics like underwater explosions (e.g. Liu et al.) and even fluid creation for visual effects in feature film (see Chapter 4...) One of this visual effects applications will be explained in more detail with the SPH software "Realfow" and NAIAD. For further reading on the applications of CFD in computer graphics please refer to [[Bridson, 2008](#)] and for new Hybrid Methods that have the advantages of Meshbased and Meshfree methods [[Jackl et al., 2006](#), pg. 195] provides a simple introduction.

5.1.4 Basic idea of SPH

SPH is a so called meshfree particle method (MPM) and replaces the meshbased discretization with a initial distribution of particles over the problem domain.

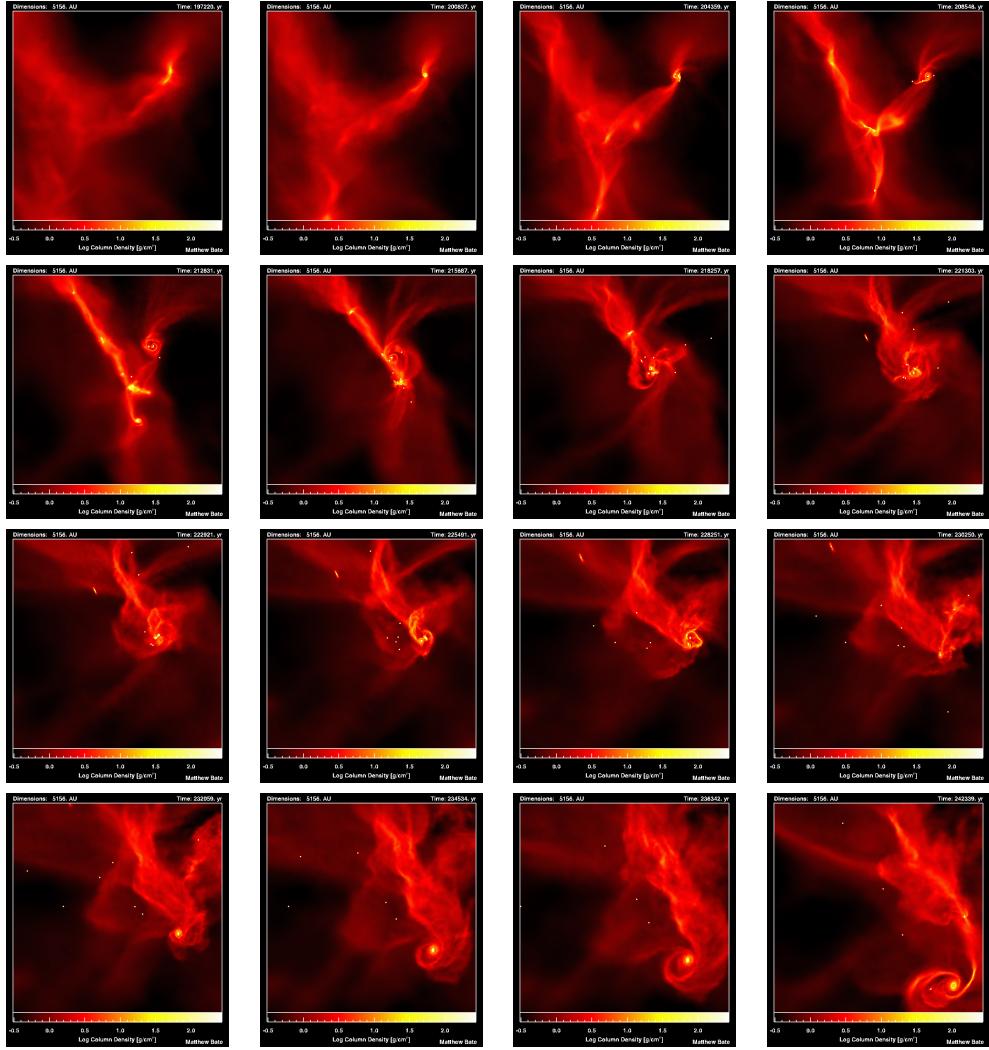


FIGURE 5.1: Image sequence of the Formation of Stars and Brown Dwarfs in a Star Cluster

5.2 Mathematical foundation

In order to discretize a domain with particles, the mathematical foundations differ greatly from Meshbased methods. Similar to FVM a integral representation of fluid variables is needed. For a continuous function $f(x)$ that has the mathematical preconditions to be represented in a integral form, it can be written in integral form as follows:

$$f(x) = \int_{\Omega} [f(a)\delta(x - a)] da \quad (x, a) \in \Omega \quad (5.1)$$

With the Dirac delta function being:

$$\delta(x - a) = \begin{cases} 1, & \text{if } x = a \\ 0, & \text{if } x \neq a \end{cases} \quad (5.2)$$

Where Ω represents the problem domain with the vectors xa in it. Because of the Dirac delta function 5.2 the integral formulation becomes zero at all vectors except the exact vector x . The form 5.1 is of great importance to SPH methods. It allows the Dirac delta function δ (which produces an exact equality) to be replaced by a *kernel function* K which leads to the approximation:

$$f(x) \approx \int_{\Omega} [f(a)K(x - a)] da \quad (x, a) \in \Omega \quad (5.3)$$

In the equations of fluid dynamics, often the divergence operator is found. The divergence of a vector function $\nabla \cdot f(x)$ can be similarly written like the above notation:

$$\nabla \cdot f(x) \approx \int_{\Omega} [\nabla \cdot f(a)K(x - a)] da \quad (x, a) \in \Omega \quad (5.4)$$

In the following section this kernel function is further discussed in brought into a form where it can be discretized.

5.2.1 Kernel approximation

The kernel function K can be rewritten with another parameter h , which as we will see is used for a similar limit formulation, as it is found in the equations of Meshbased methods. Furthermore due to already established conventions in SPH, the bracket operator $<>$ is used to clearly visualize that function $f(x)$ is approximated by the kernel function K :

$$< f(x) > = \int_{\Omega} [f(a)K(x - a, h)] da \quad (x, a) \in \Omega \quad (5.5)$$

The bracket operator can be also used to rewrite equation 5.4:

$$< \nabla \cdot f(x) > = \int_{\Omega} [\nabla \cdot f(a)K(x - a, h)] da \quad (x, a) \in \Omega \quad (5.6)$$

The relation between the Dirac delta function and the kernel function K can be written down with the limit notation by:

$$\lim_{h \rightarrow 0} K(x - a, h) = \delta(x - a) \quad (5.7)$$

Similar to equation 4.1 in FDM, the value of h is used to approximate a function. In contrast to FDM though, not the function itself that should be discretized is approximated, but the Dirac function. Equation 5.7 represent the *Delta function condition*, which is one out of three conditions that has to be met by the kernel function K , in order to be used with SPH. For further reading on the *unity condition* and the *compact condition* please refer to [Liu and Liu, 2003, pg. 37]. From now on we will call h the kernel *length*. In literature the term *kernel* is often used in exchange with the term *smoothing*, which result in smoothing function K and smoothing length h . In this report the term kernel will be used. The problem domain Ω is also called *support domain*.¹

The equation 5.8 can be rewritten to:

$$\langle \nabla \cdot f(x) \rangle = \overbrace{\int_{\Omega} \nabla \cdot [f(a)K(x - a, h)] da}^{\text{Divergence theorem applicable}} - \int_{\Omega} [f(a) \cdot \nabla K(x - a, h)] da \quad (5.8)$$

This form allows to apply the *Divergence theorem*, onto the RHS of the equation:

$$\langle \nabla \cdot f(x) \rangle = \int_A [f(a)K(x - a, h) \cdot \vec{n}] dA - \int_{\Omega} [f(a) \cdot \nabla K(x - a, h)] da \quad (5.9)$$

For vectors a which are not identical to vector x , and for which the distance between them bypasses the kernel length $|x - a| > h$ the kernel function becomes zero. This leads the integral of the surface A which is located inside the problem domain Ω to be zero. Therefore the equation simplifies to:

$$\langle \nabla \cdot f(x) \rangle = - \int_{\Omega} [f(a) \cdot \nabla K(x - a, h)] da \quad (5.10)$$

Considering the conversion of equation 5.4 to 5.10 with the help of the divergence theorem, something remarkable has happened. The divergence operation on the vector function $f(x)$ has been shifted to be a gradient operation at the kernel function K , leaving the vector function untouched. Therefore the kernel approximation in SPH which was used to represent the field function f in a integral form has the following advantage: When a divergence operation occurs at the field function (which often happens in CFD), the solution can be computed by the values of the function itself and the gradient of the

¹This is a simplification for this report. Strictly speaking the support domain can also be outside of the problem domain and therefore not equivalent.

kernel function, without caring about the divergence of the function. This will lead to much simpler and faster solvable equations. Now that the continuous integral forms are written in the kernel approximation form, in the next step the equations 5.5 and 5.10 can be finally discretized using particles.

5.2.2 Particle approximation

In order to discretize the continuous integral form that are represented by the kernel approximations, the SPH method uses particle summations. In the beginning of the process the domain is discretized or better split up into randomly distributed system of particles. These particles carry certain values like mass and density. The problem of correctly distributing the particles is further discussed in the Implementation sections of this report. In discretized form the particles represent the overall domain Ω . To compute the physical values at a certain point (or particle) inside the domain, the surrounding particles are considered and a weighted average value is computed. This entire process will be discussed in this Section.

For simplicity and better understanding we first assume a already discretized domain with randomly distributed particles. Figure 5.2 illustrates the two dimensional distribution of these particles in the domain Ω . The region from which field information should be computed is marked as the red particle i . This particle is surrounded by a *support domain* (dotted quadratic line) that defines which particles have influence on the field variable values it inhibits. This support domain is controlled by the kernel function K and its kernel length h . In the illustration the support domain is simplified as a two dimensional area, but it can also be a spheric volume with radius h or of any other shape that depends on the kernel length. The magnitude of influence is illustrated by a bell shaped kernel function, that clearly decreases with increasing distance from the particle i . Now to transform the idea of this illustration into the mathematical particle approximation of SPH, we consider the support domain to be a cubic volume V with side length h .

The domain Ω is discretized into a finite amount of N particles. Each particle j is assigned a mass m_j and a density ρ_j , which are equal fractions of the total mass and density of Ω . The mass of particles m_j inside the cubic support volume V_j can therefore be written as:

$$m_j = V_j \rho_j \quad j \in V_j \quad (5.11)$$

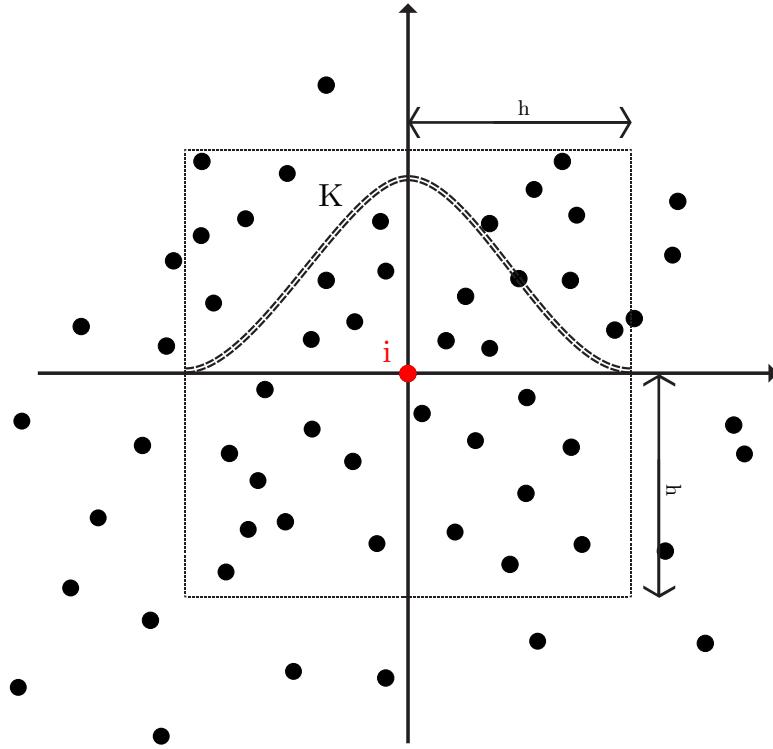


FIGURE 5.2: Illustration of particle approximation with kernel function K and quadratic support domain

Considering the kernel approximation equation 5.5, the integral over the continuous domain Ω can now be rewritten to a discrete summation:

$$\begin{aligned} \langle f(x_i) \rangle &= \int_{\Omega} [f(a)K(x-a, h)] da \\ &\approx \sum_{j=1}^N [f(a_j)K(x_i - a_j, h)V_j] \end{aligned} \tag{5.12}$$

In equation 5.12 the step from kernel approximation to particle approximation has been made. Now the vectors x_i , a_j represent particle positions. Thereby the function is approximated at particle i . This equation can be further written with a rearranged equation 5.11 in mind to:

$$\begin{aligned}
\langle f(x_i) \rangle &\approx \sum_{j=1}^N \left[f(a_j) K(x_i - a_j, h) V_j \right] \\
&= \sum_{j=1}^N \left[f(a_j) K(x_i - a_j, h) \frac{m_j}{\rho_j} \right] \\
&= \sum_{j=1}^N \left[\frac{m_j}{\rho_j} f(a_j) K(x_i - a_j, h) \right]
\end{aligned} \tag{5.13}$$

Note that in equation 5.13 the only changing input values are the particle properties (e.g. position, mass, density, ...) of the the particle(s) i, j . Furthermore the most significantly varying element in the equation is the kernel function K itself. Most SPH methods can be clearly identified by the kernel function they use to weight the influence of surrounding particles. Therefore the equation can become cleaner by separating the kernel function $K_{i,j}$ from the description:

$$\langle f(x_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f(a_j) \cdot K_{i,j} \tag{5.14}$$

Similar to the kernel approximation in equation 5.5 the particle approximation can be applied to 5.10, which finally leads to:

$$\langle \nabla \cdot f(x) \rangle = - \sum_{j=1}^N \frac{m_j}{\rho_j} f(a_j) \cdot \nabla K_{i,j} \tag{5.15}$$

In this Section the kernel approximations were discretized to the particle approximations. These two approximations and the step to latter are the mathematical cornerstone of the SPH method. The continuous integrals from the kernel approximations were transformed to summations over a finite amount of particles inside a finite support domain. This particle approximation is done at each time step of the simulation. Note that this simple idea of a averaged summations with particles was done completely without the presence of a mesh. Each particle was associated with at least properties for position, mass and density. As we will see in later Sections, further physical properties can be associated with particles for hydrodynamic problems. How the distribution of particles is done, will be discussed in later Sections. In the following Sections these mathematical framework is used to implement the NSE with SPH. Hereby to consider are algorithms to efficiently determine, which particles are inside the support domain.

5.2.3 Lagrangian Navier-Stokes equations

Because SPH is a pure Lagrangian method, also the NSE need to be formulated in the Lagrangian form. In the previous Chapters mostly the Eulerian forms were used.

5.2.4 SPH vs. FEM

While both methods in most implementations share a Lagrangian perspective the main difference between the FEM and particle based methods is indeed the absence of a overall grid in the SPH methods. While FEM is based on finite elements, they still exist within a grid.

For further reading please refer to [Li and Liu, 2004] and [Liu and Liu, 2003].

5.2.5 Benefits of Meshfree methods over Meshbased methods

The most problems of the Meshbased methods result not from the numerical capabilities of their methods, but from the simple presence of a mesh itself.

item Be@tter to model phycal behaviour that has strong distortion item Mesh dependend problems like remeshing are avoided item No tie-consuming (re-)generation of meshes for a problem

5.3 Implementation

5.3.1 Nearest neighbour particles

5.3.2 Issues of computer implementation

5.3.3 Parcallel Coding

Fitc

5.4 Implementation of Navier-Stokes equations

5.5 Particle consistency problem

Chapter 6

Useage in Simulation Software

6.1 Mesh-free methods

6.1.1 Why use mesh free methods?

6.2 Implementation

Bibliography

- J. Anderson. *Fundamentals of Aerodynamics*. Anderson series. McGraw-Hill Education, 2010. ISBN 9780073398105.
- R. Avrs. *Vectors, Tensors, and the Basic Equations of Fluid Mechaniscs*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 1989. ISBN 9780486661100.
- P.D. Bates, S.N. Lane, and R.I. Ferguson. *Computational Fluid Dynamics: Applications in Environmental Hydraulics*. Wiley, 2005. ISBN 9780470015186.
- A. Bennett. *Lagrangian Fluid Dynamics*. Cambridge Monographs on Mechanics. Cambridge University Press, 2006. ISBN 9780521853101.
- S. Biringen and C.Y. Chow. *An Introduction to Computational Fluid Mechanics by Example*. Wiley, 2011. ISBN 9780470915172.
- R. Bridson. *Fluid Simulation For Computer Graphics*. Ak Peters Series. A K PETERS Limited (MA), 2008. ISBN 9781568813264.
- T. Cebeci. *Computational Fluid Dynamics for Engineers: From Panel to Navier-Stokes Methods with Computer Programs*. Springer, 2005. ISBN 9783540244516.
- A.W. Date. *Introduction to Computational Fluid Dynamics*. Cambridge University Press, 2005. ISBN 9780521853262.
- R.L. Elder, A. Tourlidakis, and M.K. Yates. *Advances of CFD in fluid machinery design*. Professional Engineering, 2003. ISBN 9781860583537.
- J.H. Ferziger and M. Perić. *Computational methods for fluid dynamics*. Springer London, Limited, 2002. ISBN 9783540420743.
- Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181:375–389, 1977.

- J.P. Goedbloed, R. Keppens, and S. Poedts. *Advanced Magnetohydrodynamics: With Applications to Laboratory and Astrophysical Plasmas*. Cambridge University Press, 2010. ISBN 9780521879576.
- J.M. Guccione, G.S. Kassab, and M.B. Ratcliffe. *Computational Cardiovascular Mechanics: Modeling and Applications in Heart Failure*. Springer US, 2009. ISBN 9781441907301.
- C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Number Bd. 1. Elsevier Science, 2007. ISBN 9780080550022.
- W.W.F. Hughes and J.A. Brighton. *Schaum's Outline of Fluid Dynamics*. Schaum's Outline Series. McGraw-Hill Education, 1999. ISBN 9780070311183.
- D. Jackl, S. Nenreither, and F. Wagner. *Methoden Der Computeranimation*. Springer London, Limited, 2006. ISBN 9783540334071.
- E. Laurien and H. Oertel. *Numerische Strömungsmechanik: Grundgleichungen - Lösungsmethoden - Softwarebeispiele*. Studium Technik. Vieweg+Teubner Verlag, 2009. ISBN 9783834805331.
- R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. ISBN 9780521009249.
- S. Li and W.K. Liu. *Meshfree Particle Methods*. Springer, 2004. ISBN 9783540222569.
- G.G.R. Liu and B. Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World Scientific Publishing Company, Incorporated, 2003. ISBN 9789812384560.
- Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977.
- H. Martin. *Numerische Strömungssimulation In Der Hydrodynamik: Grundlagen Und Methoden*. Springer, 2011. ISBN 9783642172083.
- K. McGuffie and A. Henderson-Sellers. *A Climate Modelling Primer*. Wiley, 2005. ISBN 9780470026410.
- L.M. Milne-Thomson. *The calculus of finite differences*. Chelsea Publishing Company, 2000. ISBN 9780821821077.
- K.A. Naugolnych and L. Ostrovsky. *Nonlinear Wave Processes in Acoustics*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1998. ISBN 9780521399845.

- W.L. Oberkampf and C.J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010. ISBN 9780521113601.
- M. Potter. *Fluid Mechanics DeMYSTiFied*. McGraw-Hill "Demystified" series. McGraw-Hill Education, 2009. ISBN 9780071626828.
- C. Pozrikidis. *Fluid dynamics*. Springer London, Limited, 2009. ISBN 9780387958712.
- W.H. Press. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN 9780521880688.
- Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003. ISBN 9780898715347.
- Y. Shi, H.W. Ge, and R.D. Reitz. *Computational Optimization of Internal Combustion Engines*. SpringerLink : Bcher. Springer London, Limited, 2011. ISBN 9780857296191.
- E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer-Verlag Berlin Heidelberg, 2009. ISBN 9783540498346.
- C. Underwood and F. Yik. *Modelling Methods for Energy in Buildings*. Wiley, 2008. ISBN 9781405153003.
- H.H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education Australia, 2007. ISBN 9780131274983.
- T.T. Warner. *Numerical Weather and Climate Prediction*. Cambridge University Press, 2010. ISBN 9780521513890.
- J.F. Wendt, J. John D. Anderson, and Von Karman Institute for Fluid Dynamics. *Computational Fluid Dynamics: An Introduction*. Von Karman Institute Book. Springer, 2009. ISBN 9783540850557.
- P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer Series in Computational Mathematics. Springer, 2009. ISBN 9783642051456.
- B.I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition: With 25 Tables*. Discretization Methods and Iterative Solvers Based on Domain Decomposition. Springer Verlag, 2001. ISBN 9783540410836.
- O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*. Elsevier Science, 2005. ISBN 9780080455594.
- O. Zikanov. *Essential Computational Fluid Dynamics*. Wiley, 2010. ISBN 9780470423295.