General Smoothed Particle Hydrodynamics

Jörg Kuhnert

Vom Fachbereich Mathematik der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation.

1. Gutachter: Prof. Dr. H. Neunzert
2. Gutachter: Prof. Dr. J.J. Monaghan

Vollzug der Promotion: 14.06.1999

D 386

Ich bedanke mich bei Herrn Prof. Helmut Neunzert für die Möglichkeit, am Institut für Techno- und Wirtschaftsmathematik zu promovieren. Kritik und weitreichendes Verständnis, wie ich sie hier vorfand, dienten meiner Arbeit in sehr fruchtbarer Weise. Das sehr spannende Thema meiner Dissertation war Ansporn, täglich einen Schritt vorwärts wagen.

Danken möchte ich vor allem Dr. Michael Junk für seine geduldige Kooperation während der ganzen Zeit meiner Promotion. Weiterhin geht mein Dank an Dr. Konrad Steiner und Dr. Dietmar Hietel für viele fruchtbare Diskussionen und die Hilfe, die mir durch sie zuteil wurde.

Mein innigster Dank aber geht an Katrin für ihre unendliche Geduld und Toleranz, an Tanja für ihre Art mich zu motivieren und an meine Familie für die wunderbare Unterstützung in jeglicher Hinsicht.

Table of Contents

Chapter 1: Introduction 1

- 1.1. Some Good Motivation for this Thesis 1
- 1.2. The Classical Idea of Smoothed Particle Hydrodynamics 2
  - 1.2.1. Conservation of Mass in the Classical SPH-method 9
  - 1.2.2. Conservation of Momentum in Classical SPH 12
  - 1.2.3. Conservation of Energy in Classical SPH 14
  - 1.2.4. Entropy in Classical SPH 16
  - 1.2.5. Complete Set of Equations for Time Evolution in Classical SPH 17
  - 1.2.6. Setting up Boundary Conditions in Classical 18
  - 1.2.7. Sketch of Some Unpleasant Approximation Properties 22
- 1.3. Conclusions 23

Chapter 2: General SPH - A New Particle Scheme for Conservation Laws 25

- 2.1. Partial Differential Equations to be Approximated 27
- 2.2. General Smoothed Particle Scheme for Conservation Laws 29
- 2.3. Conservation Properties of the Numerical Scheme 31
  - 2.3.1. Conservation of Mass, Momentum, and Total Energy 31
  - 2.3.2. Conservation of the Global Error 34
  - 2.3.3. Source of Instabilities Due To Numerical Scheme 36
- 2.4. Time Integration of the General Smoothed Particle Scheme 38
  - 2.4.1. First Order Time Integration Methods 38
  - 2.4.2. Second Order Time Integration Methods 46
  - 2.4.3. A Time Integration Method for Low Mach Number Flow 47

Chapter 3: Exhibition of a Class of Smoothed Particle Operators 51

- 3.1. Construction of Smoothed Particle Operators of Different Orders of Accuracy 51
- 3.2. Practical Computation of Smoothed Particle Approximations 60
- 3.3. Practical Computation of First Derivatives of Smoothed Particle Approximations 62
- 3.4. Practical Computation of Second Derivatives of Smoothed Particle Approximations 64

Chapter 4: Handling Boundary Conditions in the General SPH Setup 65

- 4.1. Characteristic Theory for Euler Equations - Theoretical

Consideration of Boundary Behavior 65
  - 4.1.1. Inflow Boundaries 72
  - 4.1.2. Outflow Boundaries 73
  - 4.1.3. Solid Wall Boundaries 73
- 4.2. Practical Setting of Boundary Conditions 74
  - 4.2.1. Boundary Treatment for Euler Equations 74
  - 4.2.2. Boundary Treatment for Navier-Stokes Equations 75
- 4.3. Treatment of Sharp Edged Solid Wall Boundaries 75

Chapter 5: Some Useful Technical Items 79
- 5.1. Physical Weight of Particles 79
- 5.2. Searching for Nearest Neighbor Particles 80
  - 5.2.1. Nearest Neighbors for Constant Smoothing Length 80
  - 5.2.2. Nearest Neighbors for Variable Smoothing Length 83
- 5.3. Adding and Removing Particles 86
  - 5.3.1. Adding Particles 89
  - 5.3.2. Removing Particles 89

Chapter 6: Upwinding the General SPH for Euler Equations 91
- 6.1. Upwind Particle Scheme based on Characteristic Theory 91
- 6.2. Upwind General SPH in Conservative Variable Formulation 95
- 6.3. Fast Upwind General SPH 97
- 6.4. An Even Faster Upwind Particle Scheme 99

Conclusion 101

Used Symbols and Abbreviations 103

Bibliography 105

Appendix: 111
- A.1. A Single Shock Solution with In-viscid Flow 112
- A.2. A Shock System in a Convergent Channel 115
- A.3. Moving Stiff Plate in a Closed Box 118
- A.4. 'Moving Nose' Problem 123
- A.5. Viscous Flow Between Two Rotating Cylinders 128
- A.6. Viscous Flow Around Cylinder 131
- A.7. A Result Related to the Airbag Problem 136

Chapter 1

1. Introduction

1.1. Some Good Motivation for this Thesis

The SPH method is a particle method which has been used for a long time. It is already a commonly used method to compute astrophysical appearances. However, it seems to have more problems to be accepted in continuum mechanics, for computing the flow of compressible fluids. While in astrophysics, no boundaries are given and consequently no numerical treatment of boundary conditions is necessary, in continuum mechanics there are always boundaries and it is essential to involve them in the SPH model. The treatment of boundary conditions represents a very large problem in the SPH setup for continuum mechanics. Several authors (Vila, Monaghan, Hartig) tried to solve this problem and came up with convenient solutions. However, the problems were not removed in general. With the present thesis, the author would like to provide a new, but very practical and general ansatz to a particle method, which allows a proper treatment of boundary conditions of any kind.

Moreover, SPH proves to have more or less inconvenient approximation properties. The approximation properties change quickly in time and space (which is due to the fact that particles move with the velocity of the fluid and thus the geometrical grid changes quickly). The general SPH ansatz provided in this thesis takes care of convenient approximation properties. The ansatz presented here is based on a least squares approximation and allows smoothed particle approximations of higher order accuracy. A similar technique like the one presented here was given already for SPH in impact mechanics by G. Dilts (see [7]).

Some research in the field of convergence, consistency and stability of the SPH method was done in [1], [16], [29].

Another very interesting SPH approach is given by H. Yserentant (see [32]). It is based on so called mass packages with finite extension in space, representing a finite contribution of mass, momentum, angular momentum, and energy.

One could raise the question what all this work (including this thesis) is done for. Obviously, there are very popular methods in continuum mechanics like finite volume methods (FVM) as like as finite element methods (FEM), based on a very profound basis. Those methods are very powerful and are practiced in industry as well as in sciences and have proved reliability as well as flexibility. So, why does one need another method for the same class of problems?

The answer is easy. The mentioned FEM or FVM are based on computationally expensive geometrical grids. For problems with unchanging geometry, the FE- of FV grid needs to be established only once. In this case, FEM or FVM prove to be very effective. Nevertheless, if geometry changes in time or free surfaces occur, then one would have to re-adapt the grid to the geometry at almost every time step, by which the performance of the method drops down considerably. However, a particle method is a more 'friendly' method for computing fluid flow with moving boundaries or free surfaces. The particles are carriers of information which move with the velocity of the fluid. Hence, particles are 'self adapting' to the geometry. It is therefore a grid free method and so, grid re-organization does not require any additional effort.

With this thesis, we would like to improve the employability of the SPH method by a little bit.

1.2. The Classical Idea of Smoothed Particle Hydrodynamics

In this section we will find an overview of the classical method of SPH (see [17],[19]). In the first part, some preparations are necessary. The real introduction of the classical SPH idea starts with section 1.2.1. on page 9. In this section there will be nothing new. However, we found it to be important to give a short review on what was done in this field up to now. In chapter 2 we will go over to our own ideas, which find their roots in the classical ansatz. However, it turns out that our newly developed method proves to have moved a little step away from the classical basis.

The classical SPH-method in continuum mechanics was developed to compute numerical solutions to time-dependent flow of compressible fluids governed by the Euler equations. The method uses particles as carriers of information characterizing the fluid flow in a certain domain $\Omega(t)$. Suppose there are N particles distributed in $\Omega(t)$. Besides the location $\mathbf{x}_i$ and the 'weight' m_i, any particle with the index i carries information like temperature, pressure, velocity; in other words all information necessary to solve the considered problem. Particle i itself moves with the same velocity which a fluid particle at the same geometrical location would move with. Considering the compressible Euler equations

$$\frac{d\rho}{dt} = -\rho \cdot \nabla \mathbf{v} \tag{1.1}$$

$$\frac{d\mathbf{v}}{dt} = -\frac{\nabla p}{\rho} \tag{1.2}$$

$$\frac{du}{dt} = -\frac{p}{\rho} \cdot \nabla \mathbf{v} \tag{1.3}$$

one notices that for any numerical computation of the time evolution of the fluid flow the computation of spatial derivatives of pressure and velocity is required. Although the values for velocity and pressure are known at the particle positions, the computation of good approximations of the derivatives is not trivial. In order to obtain good approximations, the classical SPH method uses the particle approximation which is described below.

Let us begin with some definitions which will be useful for setting up the classical SPH method.

Definition 1.1:
Let $\Omega(t) \subset \mathbf{R}^\nu$ be measurable. Let $N \in \mathbf{N}$.
1.) The set of pairs

$$\omega_N(t) = \{(m_i, \mathbf{x}_i(t)); i = 1...N; \mathbf{x}_i(t) \in \Omega(t)\} \tag{1.4}$$

is called an ensemble of particles or particle system. We call $\mathbf{x}_i$ the position of the particles with index i and m_i the corresponding weight. $\Omega(t)$ is the domain of fluid flow.
2.) By $\mathcal{F}(\Omega(t))$ we denote the set of measurable densities on $\Omega(t)$

$$\mathcal{F}(\Omega(t)) := \left\{ f : \Omega(t) \to \mathbf{R}^+; \quad \int_{\Omega(t)} f d\mathbf{x} \quad \text{exists} \right\}$$

where the integral is taken in the Lebesgue sense.
3.) We define the set $\bar{\Omega}$ by

$$\bar{\Omega} := \{ \ (t, \mathbf{x}) \mid t \in T \ , \ \mathbf{x} \in \Omega(t) \ \}$$

The following definition gives a measure on $\Omega(t)$

Definition 1.2:
Let $\rho(t) \in \mathcal{F}(\Omega(t))$ be a density, i.e. $\rho(t) \geq 0$ a.e., $\rho(t) \in \mathbf{L}^1(\Omega(t))$. We define the measure $\mu_\rho(t, V)$ by

$$\mu_\rho(t, V) = \int_V \rho(t) \, d\mathbf{x} \qquad \forall \text{ measurable sets } V \subset \Omega(t) \tag{1.5}$$

and call it the measure corresponding to $\rho(t)$. $\mu_\rho(t, V)$ is the total mass in V provided by $\rho(t)$ at some time t.

Definition 1.3:
Let $\rho(t) \in \mathcal{F}(\Omega(t))$ and μ_ρ be its corresponding measure. Let $\phi : \bar{\Omega} \to \mathbf{R}$ be a smooth function. Let $V \subset \Omega(t)$ be measurable. We call the measure defined by

$$\mathcal{M}_\rho[\phi](t, V) := \int_V \phi(t) \, d\mu_\rho \tag{1.6}$$

the $\mathcal{M}$-measure of ϕ with respect to the measure μ_ρ.

Now, one could go ahead and try to approximate functions by the help of the defined $\mathcal{M}$-measure. From the point of view of fluid flow, the total mass in V would be $\mathcal{M}_\rho[1](t, V)$, where ρ is the density of the fluid. In the same way, the total momentum in V would be $\mathcal{M}_\rho[\mathbf{v}](t, V)$, where $\mathbf{v}$ is the velocity of the fluid. However, it shall not be possible to evaluate the $\mathcal{M}$-measure, since in the sense of setting up a particle method, quantities of density, velocity and energy are known at the $\mathbf{x}_i$ only and, hence, the evaluation of the given integrals is impossible. Thus, it is necessary to come up with a suitable measure which at least approximates the $\mathcal{M}$-measure.

Definition 1.4:
Let $\omega_N(t)$ be an ensemble of particles. We define by

$$\delta_{\omega_N}(t, V) := \int_V m_i \, \delta(\mathbf{x} - \mathbf{x}_i) \, d\mathbf{x} = \sum_{i:\mathbf{x}_i(t) \in V} m_i \tag{1.7}$$

a discrete measure on $\Omega(t)$ and call it particle measure.

By use of the discrete measure δ_{ω_N} we are in the situation to define discrete moments (similar to definition 1.3):

Definition 1.5:
Let $\omega_N(t)$ be a particle ensemble as defined above. Let $\phi : \bar{\Omega} \to \mathbf{R}$. Let $V \subset \Omega(t)$

be measurable. The measure defined by

$$M_\delta[\phi](t,V) := \int_V \phi \, d\delta_{\omega_N} = \sum_{i:\mathbf{x}_i(t)\in V} \phi(t,\mathbf{x}_i(t)) \cdot m_i \tag{1.8}$$

is called the M-measure of ϕ with respect to the particle measure.

We now try to use δ_{ω_N} to approximate μ_ρ because if we succeed to do so then it would be possible to give approximations of the $\mathcal{M}$-measure by using the M-measure of ϕ in the sense that

$$\begin{aligned} \mathcal{M}_\rho[\phi](t,V) &= && (1.9)\\ \int_V \phi \, d\mu_\rho &\approx && (1.10)\\ \int_V \phi \, d\delta_{\omega_N} &= && (1.11)\\ \sum_{i:\mathbf{x}_i(t)\in V} \phi(t,\mathbf{x}_i(t)) \cdot m_i &= M_\delta[\phi](t,V) && (1.12) \end{aligned}$$

Suppose the discrete measure δ_{ω_N} approximates the continuous measure μ_ρ, then we would like to introduce a distance between the two measures which tells, how 'close' the two are together. There are many choices i.e. metrics in metric spaces; we choose

Definition 1.6:
Given a continuous and a discrete measures μ_ρ and δ_{ω_N} as defined above. Let $V \subset \Omega(t)$ be rectangular. We define by

$$D(\delta_{\omega_N}, \mu_\rho) := \sup_{V\subset\Omega(t)} (| \int_V d\delta_{\omega_N} \quad - \quad \int_V d\mu_\rho |) \tag{1.13}$$

the distance between the two measures.
Remark: The distance is indeed dependent on time t because $\omega_N(t)$ and also $\rho(t)$ change in time.

The Lemma of Koksma and Hlawka now closes the circle by showing, how well the continuous moment of a function $\phi(t)$ is approximated by its discrete moment.

Lemma 1.7 (Koksma-Hlawka)
Let $\omega_N(t)$ be a particle ensemble, $\delta_{\omega_N}(t)$ be the corresponding discrete measure. Let $\rho(t) \in \mathcal{F}(\,\Omega(t)\,)$ and $\mu_\rho(t)$ be the corresponding measure. Let $\phi : \bar{\Omega} \to \mathbf{R}$ be a

function of bounded variation. Then we have for any measurable $V \subset \Omega(t)$

$$|\mathcal{M}_\rho[\phi](t,V) \quad - \quad M_\delta[\phi](t,V)| \quad \leq \quad var(\phi(t)) \cdot D(\delta_{\omega_N}, \mu_\rho) \tag{1.14}$$

where $var(\phi(t))$ denotes the variation of the function $\phi(t)$ in the sense of Hardy and Krause.

Now, we are in the situation to give approximations of $\mathcal{M}_\rho[\phi](t,V)$, provided that we established a particle ensemble $\omega_N(t)$ with possibly small distance $D(\delta_{\omega_N}, \mu_\rho)$. This distance, however, depends only on the choice of the $\mathbf{x}_i$ and m_i. Thus it is up to the user to establish a particle ensemble with conveniently small distance between the two measures at any time $\bar{t}$. Practically, this can be done in the three following ways.

- At some time $\bar{t}$, for a given density distribution $\rho(\bar{t})$, we distribute the particles on a regular mesh in $\Omega(\bar{t})$ and then try to find the weights m_i such that the distance is small.
- At some time $\bar{t}$, for a given density distribution $\rho(\bar{t})$, we distribute the weights to the particles and then the task is to find out the $\mathbf{x}_i$ such that $D(\delta_{\omega_N}, \mu_\rho)$ is small.
- At some time $\bar{t}$, for given weights and positions, we try to establish a density $\rho(\bar{t})$ such that the distance is small.

The classical SPH-method is based on the theory given above. $\Omega(t)$ denotes the domain of fluid flow. In the following we would like to show, how fluid flow is computed in the classical SPH sense.

The SPH setup at some initial time t_0 is given by

- a scalar function $\rho(t_0) \in \mathcal{F}(\,\Omega(t_0)\,)$ describing the initial mass density of the fluid and $\mu_\rho(t_0)$ the corresponding measure
- a vector valued function $\mathbf{v}(t_0) : \Omega(t_0) \to \mathbf{R}^\nu$ describing the initial velocity of the fluid
- a scalar function $u(t_0) : \Omega(t_0) \to \mathbf{R}^\nu$ describing the internal energy (in thermodynamical sense) of the fluid
- a particle ensemble $\omega_N(t_0)$ with conveniently small $D(\delta_{\omega_N}, \mu_\rho)$ is established in $\Omega(t_0)$
- a family of smoothing kernels $\{\, W_h : \mathbf{R}^\nu \to \mathbf{R} | h > 0 \,\}$ is defined which provides the following properties

- $W_h(\xi) \geq 0$ if $\|\xi\|_2 \leq h$, $W_h(\xi) = 0$ if $\|\xi\|_2 \leq h$
- $\int_{\mathbf{R}^\nu} W_h(\xi)\, d\xi = 1$
- W_h is point-symmetric in 0
- W_h is smooth

The parameter h in the definition of the smoothing kernel W_h is called smoothing length and represents the maximum interaction radius between two particles.

In the following sections 1.2.1, 1.2.2, and 1.2.3, we would like to outline, how in classical SPH particles move and how quantities like density, momentum and energy are updated at any time $t > t_0$. Before we turn to these considerations we have to give three preparing definitions.

Definition 1.8:
A complete particle system is given by the collection of all information necessary to describe the considered problem of fluid flow

$$\omega_N^c(t) = \{\ (m_i, \mathbf{x}_i(t), \mathbf{y}_i(t))\ ;\ i = 1...N;\ \mathbf{x}_i(t) \in \Omega(t)\ \} \tag{1.15}$$

In addition to definition 1.1 we call $\mathbf{y}_i(t)$ the vector of necessary information to completely describe the considered problem in fluid flow. For the case of Euler flow, we would have $\mathbf{y}_i(t) = (\mathbf{v}_i^T(t),\ u_i(t))$ where $\mathbf{v}_i(t)$ is the velocity of the fluid and $u_i(t)$ the internal energy of the fluid at particle i.

For the reason of simplification, it is favorable to define special symbols for continuous and discrete moments of special functions. Hence, the definition below specifies those special moments which are needed by the classical SPH method.

Definition 1.9:
Given $\rho(t) \in \mathcal{F}(\Omega(t))$. Let W_h be a smoothing kernel. Given a smooth function $f : \bar{\Omega} \to \mathbf{R}$. Then we denote the convolution

$$\pi f(t, \mathbf{y}) := \int_\Omega W_h(\mathbf{y} - \mathbf{x}) \cdot f(t, \mathbf{x})\, d\mathbf{x} = \mathcal{M}_\rho \left[W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\rho} \right] (t, \Omega(t)) \tag{1.16}$$

and equivalently the gradient of this function

$$\nabla \pi f(t, \mathbf{y}) = \int_\Omega \nabla W_h(\mathbf{y} - \mathbf{x}) \cdot f(t, \mathbf{x})\, d\mathbf{x} = \mathcal{M}_\rho \left[\nabla W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\rho} \right] (t, \Omega(t)) \tag{1.17}$$

We call πf the smoothed approximation (or simply π-approximations) of f.

Definition 1.10:
Let $\omega_N(t)$ be a particle ensemble, W_h a smoothing kernel. Given a smooth function $f : \bar{\Omega} \to \mathbf{R}$. The smooth functions

$$\begin{aligned}\rho_\Pi(t, \mathbf{y}) &:= \sum_{i=1}^{N} W_h(\mathbf{y} - \mathbf{x}_i(t))\, m_i = M_\delta[W_h(\mathbf{y} - \cdot)](t, \Omega(t)) && (1.18)\\ \Pi f(t, \mathbf{y}) &:= \sum_{i=1}^{N} W_h(\mathbf{y} - \mathbf{x}_i(t)) \frac{f(t, \mathbf{x}_i)}{\rho_\Pi(t, \mathbf{x}_i)}\, m_i \\ &= M_\delta\left[W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\rho_\Pi}\right](t, \Omega(t)) && (1.19)\end{aligned}$$

are called smoothed particle approximations (or simply Π-approximations) of the functions ρ and f respectively. We denote by

$$\begin{aligned}\nabla(\ \rho_\Pi(t, \mathbf{y})\ ) &= \sum_{i=1}^{N} \nabla W_h(\mathbf{y} - \mathbf{x}_i(t))\, m_i = M_\delta[\nabla W_h(\mathbf{y} - \cdot)](t, \Omega(t)) && (1.20)\\ \nabla(\ \Pi f(t, \mathbf{y})\ ) &= \sum_{i=1}^{N} \nabla W_h(\mathbf{y} - \mathbf{x}_i(t)) \frac{f(t, \mathbf{x}_i)}{\rho_\Pi(t, \mathbf{x}_i)}\, m_i \\ &= M_\delta\left[\nabla W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\Pi\rho}\right](t, \Omega(t)) && (1.21)\end{aligned}$$

the gradients of the smoothed particle approximations.

Lemma 1.11: (Approximation properties of the π-approximation)
Given a smoothing kernel W_h as defined above. Given a smooth function $f \in \bar{\Omega} \to \mathbf{R}$ and its corresponding convolution $\pi f(t)$. Then the following holds:

$$\begin{aligned}|\ \pi f(t, \mathbf{y}) - \quad f(t, \mathbf{y})\ | &= \mathbf{O}(h^2) && (1.22)\\ \|\ \nabla \pi f(t, \mathbf{y}) \quad - \quad \nabla f(t, \mathbf{y})\ \| &= \mathbf{O}(h^2) && (1.23)\end{aligned}$$

Proof:
The proof uses Taylor series expansion of the function f.

By the help of lemma 1.7 and lemma 1.11 we can state

Lemma 1.12: (Approximation properties of the Π-approximation)
Let $\Pi\rho$, πf, and Πf be as defined above. Then we have

$\Pi f(t, \mathbf{y})$ is an approximation of $\pi f(t, \mathbf{y})$ in the sense

$$|\pi f(t, \mathbf{y}) - \Pi f(t, \mathbf{y})| \leq var(W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\rho_\Pi}) \cdot D(\delta_{\omega_N}, \mu_{\rho_\Pi}) \tag{1.24}$$

$$\|\nabla \pi f(t, \mathbf{y}) - \nabla \Pi f(t, \mathbf{y})\| \leq var(\nabla W_h(\mathbf{y} - \cdot) \cdot \frac{f}{\rho_\Pi}) \cdot D(\delta_{\omega_N}, \mu_{\rho_\Pi}) \tag{1.25}$$

Proof:
The proof combines lemmas 1.7 and lemma 1.11.
Remark: Combining lemmas 1.11 and 1.12, we recognize that the approximation quality of Πf depends on the sizes of distance D and smoothing length h.

We did not mention yet, how large $D(\delta_{\omega_N}, \mu_{\rho_\Pi})$ can grow. It mainly depends on the particle distribution. Nothing is known about this, since it certainly depends on the distribution of the particles. We cannot give safe statements on local approximation properties. However, we would like to outline a very interesting global approximation property shown in the small computation below.

$$\begin{aligned}
\int_{\mathbf{R}^\nu} d\mu_{\rho_\Pi} &= \int_{\mathbf{R}^\nu} \Pi\rho \, d\mathbf{x} = \int_{\mathbf{R}^\nu} \left\{ \sum_{i=1}^{N} W_h(\mathbf{x} - \mathbf{x}_i) m_i \right\} d\mathbf{x} \\
&= \sum_{i=1}^{N} \left\{ \int_{\mathbf{R}^\nu} W_h(\mathbf{x} - \mathbf{x}_i) \, d\mathbf{x} \right\} m_i = \sum_{i=1}^{N} m_i \\
&= \int_{\mathbf{R}^\nu} d\delta_{\omega_N} \qquad (1.26)
\end{aligned}$$

In the next sections we will find that exactly this global approximation behavior will provide some interesting properties of the classical SPH scheme.

Suppose at the initial time t_0, a complete particle system $\omega_N^c(t_0)$ is given. The aim of SPH is to evaluate the particle system $\omega_N^c(t)$ at a later time t. The particles are moved with the velocity which a fluid particle at the same point would move with. Thus it is left to establish rules for $\frac{d\mathbf{y}_i}{dt}$ such that the time dependent fluid flow in $\Omega(t)$ is well approximated. In the following sections 1.2.1, 1.2.2, 1.2.3, and 1.2.4 we will answer the question how well classical SPH approximates real fluid flow governed by the Euler equations.

1.2.1. Conservation of Mass in the Classical SPH-method

As we mentioned previously, classical SPH was established to simulate in-stationary, compressible fluid flow. Concerning a compressible flow, the density of the fluid is not constant, but may change in space and time. This change is described by equation (1.1), which does not mean anything else than that the mass of any arbitrary lump of fluid is conserved. In classical SPH, the approximation function of the density is ρ_Π. It will have to change approximately as the original density ρ changes. In the following we will see that, if the particles are moved with the velocity of the fluid, then the time change of $\Pi\rho$ is an approximation of the time change of the original density ρ. For the reason of showing approximation properties of ρ_Π, we rewrite (1.1) as

$$\frac{d\rho}{dt} = \nabla\rho \cdot \mathbf{v} - \nabla(\rho\mathbf{v}) \tag{1.27}$$

Lemma 1.13:
Let $\Omega(t)$ be the domain of fluid flow with neither inflow nor outflow, i.e. no mass is lost over the boundary. All particles remain inside of $\Omega(t)$. Let $\rho \in \mathcal{F}(\Omega(t_0))$ be the mass density of the fluid at the initial time t_0 and μ_ρ be its corresponding measure. Let $\omega_N(t_0)$ be a particle ensemble established in $\Omega(t_0)$ and $\delta_{\omega_N}(t_0)$ the corresponding particle measure providing a conveniently small distance $D(\delta_{\omega_N}, \mu_\rho)$. Let $\mathbf{v} : \bar{\Omega} \to \mathbf{R}^\nu$ be the (given) velocity of the fluid. Let the particles move with the velocity $\mathbf{v}$, i.e. $\frac{d}{dt}\mathbf{x}_i := \mathbf{v}(t, \mathbf{x}_i)$; let the m_i remain constant, i.e. $\frac{d}{dt}m_i = 0$. Then we can establish the following statements
1.) The discrete conservation of mass

$$\frac{d}{dt}M_\delta[1](t,\Omega) = 0 \tag{1.28}$$

holds.

2.) The local evolution of the smoothed density is given by

$$\frac{d\rho_\Pi}{dt} = \nabla\rho_\Pi \cdot \mathbf{v} - \nabla\Pi(\rho_\Pi\mathbf{v}) \tag{1.29}$$

Proof:
1.) For $M_\delta[1](t,\Omega)$ we can write

$$M_\delta[1](t,\Omega) = \sum_{i=1}^{N} m_i$$

Since we leave the m_i unchanged and also the number of particles in $\Omega(t)$ does not change by assumption, it follows that

$$\frac{d}{dt}M_\delta[1](t,\Omega) = \sum_{i=1}^{N} \frac{dm_i}{dt} = 0 \tag{1.30}$$

which concludes the first part of the proof.
Remark: As we saw in Lemma 1.7, $M_\delta[1](t,\Omega)$ is an approximation of $\mathcal{M}_\rho[1](t,\Omega)$. However, $\mathcal{M}_\rho[1](t,\Omega)$ itself represents the total mass of the fluid in the domain $\Omega(t)$, however by the assumptions of this lemma we have $\frac{d}{dt}\mathcal{M}_\rho[1](t,\Omega) = 0$. Thus, the statement $\frac{d}{dt}M_\delta[1](t,\Omega) = 0$ reveals good similarity to the statement $\frac{d}{dt}\mathcal{M}_\rho[1](t,\Omega) = 0$, meaning that no discrete mass is lost.

2.) We compute the total time derivative of $\rho_\Pi(t,\mathbf{x})$ and obtain

$$\begin{aligned}
\frac{d\rho_\Pi}{dt}(t,\mathbf{y}) &= \frac{d}{dt}\sum_{i=1}^{N} W_h(\mathbf{y}-\mathbf{x}_i)\cdot m_i \qquad (1.31)\\
&= \sum_{i=1}^{N} \nabla W_h(\mathbf{y}-\mathbf{x}_i)\cdot\left(\frac{d\mathbf{y}}{dt}-\frac{d\mathbf{x}_i}{dt}\right)\cdot m_i\\
&= \sum_{i=1}^{N} \nabla W_h(\mathbf{y}-\mathbf{x}_i)\cdot(\mathbf{v}(t,\mathbf{y})-\mathbf{v}(t,\mathbf{x}_i))\cdot m_i\\
&= \left(\sum_{i=1}^{N} \nabla W_h(\mathbf{y}-\mathbf{x}_i)\cdot m_i\right)\cdot\mathbf{v}(t,\mathbf{y}) - \\
&\qquad\qquad \sum_{i=1}^{N} \nabla W_h(\mathbf{y}-\mathbf{x}_i)\cdot\mathbf{v}(t,\mathbf{x}_i)\cdot m_i\\
&= \nabla\rho_\Pi\cdot\mathbf{v} - \nabla\Pi(\rho_\Pi\mathbf{v}) \qquad (1.32)
\end{aligned}$$

which concludes the second part of the proof.
Remark: The second part of this lemma demonstrates that not only mass is conserved globally (in a discrete sense) but also that $\frac{d\Pi\rho}{dt}$ is close to $\frac{d\rho}{dt}$ provided that $\Pi\rho$ and $\Pi(\rho\mathbf{v})$ are good approximations of ρ and $\rho\mathbf{v}$. Equation (1.29) reveals good similarity to equation (1.27).

Up to now, we showed approximation properties of equation (1.1). However, in continuum mechanics, we need to satisfy not only the law of conservation of mass, but we also have to be concerned about the laws of conservation of momentum and conservation of total energy (equations (1.2) and (1.3)). Fortunately, the classical SPH provides a very practical method to conserve momentum and energy in a discrete sense by using smoothed particle approximations as they were introduced above.

1.2.2. Conservation of Momentum in Classical SPH

The law of conservation of momentum plays an important role in computational continuum mechanics. In differential form, it is stated in equation (1.2). Nevertheless, it has a well known background, since it is the application of Newtons second law ($F = a \cdot m$) to the conditions of a moving fluid. It rules the equilibrium of forces in the fluid and it was therefore an important task to provide good approximations of (1.2) in the classical SPH scheme. In order to come up with a good solution to this, in classical SPH we rewrite equation (1.2) as

$$\frac{d\mathbf{v}}{dt} = -\nabla \left(\frac{p}{\rho}\right) - \frac{p}{\rho^2}\,\nabla\rho \tag{1.33}$$

We use now the smoothed particle approximation to define the time evolution of the velocity of particle j according to equation (1.33).

Definition 1.14:
Given a particle ensemble $\omega_N(t_0)$ at an initial time t_0, given a mass density distribution $\rho(t) \in \mathcal{F}(\Omega(t))$ and a pressure distribution $p : \bar{\Omega} \to \mathbf{R}$. Furthermore, let a smooth initial velocity distribution $\mathbf{v}_0 : \Omega(t_0) \to \mathbf{R}^\nu$ be given. Let particles i move with $\mathbf{v}_i$, i.e. $\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i$. Then we define the time evolution of the velocity of particle j by

$$\frac{d\mathbf{v}_j}{dt} = -\nabla\Pi\left(\frac{p}{\rho_\Pi}\right)(t, \mathbf{x}_j) - \frac{p}{\rho_\Pi^2}\,\nabla(\rho_\Pi)(t, \mathbf{x}_j) \tag{1.34}$$

with initial condition

$$\mathbf{v}_i(t_0) = \mathbf{v}_0(\mathbf{x}_i(t_0))$$

By virtue of lemmas 1.11 and 1.12 we observe immediately that (1.34) is an approximation of the law of conservation of momentum (1.33).

However, the local approximation properties of (1.34) depend on $D(\delta_{\omega_N}, \mu_{\rho_\Pi})$, and D might change rapidly as particles move. So there is no guaranty about certain local approximation properties. Nevertheless, we can show, that the discrete momentum of the global flow domain $\Omega(t)$ is conserved no matter what disorder the particles appear with.

Lemma 1.15:
Given the time evolution of the velocity of the particle ensemble as defined in definition 1.14. Then the discrete conservation of momentum

$$\frac{d}{dt} M_\delta[\mathbf{v}](t, \Omega) = \mathbf{0} \tag{1.35}$$

holds.

Proof:
The proof is to be found (for example) in [19] and is redrawn below.
Since the m_j are left constant, the left hand side of equation (1.35) appears as

$$\frac{d}{dt} M_\delta[\mathbf{v}](t, \Omega) = \frac{d}{dt} \sum_{j=1}^{N} \mathbf{v}_j \cdot m_j = \sum_{j=1}^{N} \frac{d\mathbf{v}_j}{dt} \cdot m_j$$

We rewrite equation (1.34) in summation form

$$\frac{d\mathbf{v}_j}{dt} = -\sum_{i=1}^{N} \left(\frac{p_i}{(\rho_\Pi)_i^2} + \frac{p_j}{(\rho_\Pi)_j^2} \right) \cdot \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \cdot m_i \tag{1.36}$$

where $(\rho_\Pi)_j := \rho_\Pi(t, \mathbf{x}_j)$

By plugging the result of equation (1.36) in, we obtain

$$\sum_{j=1}^{N} \frac{d\mathbf{v}_j}{dt} \cdot m_j = -\sum_{j=1}^{N} \sum_{i=1}^{N} \left(\frac{p_i}{(\rho_\Pi)_i^2} + \frac{p_j}{(\rho_\Pi)_j^2} \right) \cdot \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \cdot m_i m_j \tag{1.37}$$

By the symmetry of W_h it follows immediately that ∇W_h is antisymmetric, hence $\nabla W_h(\mathbf{x}_j - \mathbf{x}_i) = -\nabla W_h(\mathbf{x}_i - \mathbf{x}_j)$. Hence, for the summation elements of the sum (1.37) we can state

$$S_{ij} = -S_{ji} \tag{1.38}$$

where

$$S_{ij} := \left(\frac{p_i}{(\rho_\Pi)_i^2} + \frac{p_j}{(\rho_\Pi)_j^2} \right) \cdot \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \cdot m_i m_j$$

Under this condition we find

$$\sum_{j=1}^{N} \sum_{i=1}^{N} S_{ij} = 0 \tag{1.39}$$

which shows (1.35) and the proof is concluded.

Lemma 1.15 shows that, if we represent the continuum by a particle ensemble $\omega_N(t)$, then each particle contributes a finite, but small, piece of momentum to the whole system. Determined amounts of momentum are interchanged between particles but no amount of momentum is lost globally. So, the discrete form of conservation of momentum in this form works only for the whole flow domain $\Omega(t)$ but not for a subset. This is a nice similarity to the physical fact, that $\frac{d}{dt}\mathcal{M}_\rho[\mathbf{v}](t,\Omega(t)) = 0$ provided that $\int_{\partial\Omega(t)} p \cdot \mathbf{n}\, d\sigma = \mathbf{0}$. In fact, in classical SPH we are not able to change momentum inside of $\Omega(t)$ unless we introduce certain mechanisms to change it. This is not trivial and will be a question of setting up boundary conditions for the classical scheme, which will briefly be discussed in section 1.2.6.

We have to say a few words about why using equation (1.33) for approximation of the law of conservation of momentum instead of equation (1.2). The reason is clear. The approximation of equation (1.2) would not lead to discrete conservation of momentum.

1.2.3. Conservation of Energy in Classical SPH

Up to now, two important conservation laws of continuum mechanics are approximated by the particle method and are satisfied in a discrete sense. Now we will turn to some considerations on the conservation of energy provided by the classical SPH method. The law of conservation of energy is given by (1.3) and may equivalently be written as

$$\frac{du}{dt} = \frac{p}{\rho^2}\left(\nabla\rho \cdot \mathbf{v} - \nabla(\rho\mathbf{v})\right) \tag{1.40}$$

Similarly to what was done for setting up a scheme for conservation of momentum, we approximate equation (1.40) by means of the previously described smoothed particle approximation.

Definition 1.16:
Given a particle ensemble $\omega_N(t)$, a velocity distribution $\mathbf{v} : \bar{\Omega} \to \mathbf{R}^\nu$, and a pressure distribution $p : \bar{\Omega} \to \mathbf{R}$. Furthermore, given an initial distribution of internal energy $u_0 : \Omega(t_0) \to \mathbf{R}$. Then we define the time evolution of the internal energy u

of particle j by

$$\frac{du_j}{dt} = \frac{p_j}{(\rho_\Pi)_j^2}(\ (\nabla\rho_\Pi)_j \cdot \mathbf{v}_j - \nabla\Pi(\rho_\Pi \mathbf{v})_j\ ) \tag{1.41}$$

where index j indicates the values at position $\mathbf{x}_j$. The initial condition is given by

$$u_j(t_0) = u_0(\mathbf{x}_j(t_0))$$

By virtue of lemmas 1.11 and 1.12, we observe again that (1.41) is an approximation of the law of conservation of energy (1.40). Similarly to lemma 1.15 where we showed that momentum is conserved in a discrete sense, we can also show, that total energy is conserved in a discrete sense when using the scheme (1.41) as approximation of the law of conservation of energy. For this reason we establish

Lemma 1.17:
Given the time evolution of the internal energy of a particle ensemble as defined in definition 1.16. Then the discrete conservation of total energy

$$\frac{d}{dt} M_\delta \left[u + \frac{1}{2}\mathbf{v}^2 \right] = 0 \tag{1.42}$$

holds.

Proof:
The proof is to be found in [19] and is redrawn below.
We rewrite the left hand side of equation (1.42) as

$$\sum_{j=1}^{N} \left(\mathbf{v}_j^T \cdot \frac{d\mathbf{v_j}}{dt} \ + \ \frac{du_j}{dt} \right) \cdot m_j = LHS \tag{1.43}$$

and substitute for $\frac{d\mathbf{v}_j}{dt}$ and $\frac{du_j}{dt}$ the appropriate expressions in summation form. We obtain:

$$\begin{aligned} \sum_{j=1}^{N} \Bigg[\ -\mathbf{v}_j^T \cdot \ \sum_{i=1}^{N} \left(\tfrac{p_i}{(\rho_\Pi)_i^2} \ + \ \tfrac{p_j}{(\rho_\Pi)_j^2} \right) \cdot \nabla W_{h,ji} \cdot m_i \ \ - \\ \tfrac{p_j}{(\rho_\Pi)_j^2} \cdot \sum_{i=1}^{N} (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{h,ji} \cdot m_i \ \Bigg] \cdot m_j \ \ = \ \ LHS \end{aligned} \tag{1.44}$$

where $\nabla W_{h,ji} := \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)$.

(1.44) may be rewritten as

$$\begin{aligned} - \sum_{j=1}^{N}\sum_{i=1}^{N} \Bigg[\left(\tfrac{p_i}{(\rho_\Pi)_i^2} + \tfrac{p_j}{(\rho_\Pi)_j^2} \right) \mathbf{v}_j^T \ + \\ \tfrac{p_j}{(\rho_\Pi)_j^2} \left(\mathbf{v}_i^T - \mathbf{v}_j^T \right) \Bigg] \cdot \nabla W_{h,ji} \cdot m_j m_j \ \ = \ \ LHS \end{aligned} \tag{1.45}$$

After simplifying we obtain

$$-\sum_{j=1}^{N}\sum_{i=1}^{N}\left[\frac{p_i}{(\rho_\Pi)_i^2}\mathbf{v}_j^T \;+\; \frac{p_j}{(\rho_\Pi)_j^2}\mathbf{v}_i^T\right]\cdot\nabla W_{h,ji}\cdot m_j m_j \quad=\quad LHS \tag{1.46}$$

Changing the role of i and j, we find

$$-\sum_{j=1}^{N}\sum_{i=1}^{N}\left[\frac{p_i}{(\rho_\Pi)_i^2}\mathbf{v}_j^T \;+\; \frac{p_j}{(\rho_\Pi)_j^2}\mathbf{v}_i^T\right]\cdot\nabla W_{h,ij}\cdot m_i m_j \quad=\quad LHS \tag{1.47}$$

Finally, by the asymmetry of ∇W_h providing the relation $W_{h,ij} = -W_{h,ji}$, it follows

$$-LHS \quad=\quad LHS \tag{1.48}$$

and we observe that

$$LHS = 0 \tag{1.49}$$

which concludes the proof.

Remark: Lemma 1.17 demonstrates a nice similarity to the physical fact that $\frac{d}{dt}\mathcal{M}_\rho\left[u+\frac{1}{2}\mathbf{v}^2\right] = 0$, provided that $\int_{\partial\Omega(t)}(p\,\mathbf{v}^T)\cdot\mathbf{n}\,d\sigma = 0$. So, each particle contributes a certain piece of total energy to the whole system. Total energy may be exchanged between particles but no total energy can be lost globally. Again, this also shows that we are requested to introduce additional algorithms into the SPH setup in order to change total energy globally (when inflow/outflow boundaries occur). See section 1.2.6. for some further information on this (not trivial) matter.

1.2.4. Entropy in Classical SPH

We have to make some considerations about the time evolution of physical entropy provided by the SPH scheme. The change of physical entropy s in the macroscopic sense is given by

$$ds = \frac{1}{T}\cdot\left(du - \frac{p}{\rho^2}\,d\rho\right) \tag{1.50}$$

(see [3]) where T denotes the absolute temperature and p denotes the pressure. This is, in some sense, an equation of state in thermodynamics telling that the entropy changes as internal energy and density change. We can now express the change of entropy of one particle of a given complete particle system $\omega_N^c(t)$ by the help of equation (1.50).

$$\frac{ds_j}{dt} = \frac{1}{T} \cdot \left(\frac{du_j}{dt} - \frac{p_j}{(\rho_\Pi)_j^2} \frac{d(\rho_\Pi)_j}{dt} \right) \tag{1.51}$$

If we replace $\frac{du_j}{dt}$ and $\frac{d(\rho_\Pi)_j}{dt}$ by the numerical expressions, we obtain

$$\begin{aligned} \frac{ds_j}{dt} &= \frac{1}{T_j} \cdot \left(\frac{p_j}{(\rho_\Pi)_j^2} (\ (\nabla \rho_\Pi)_j \ \cdot \mathbf{v}_j - \nabla \Pi (\rho_\Pi \mathbf{v})_j \ ) - \right. \\ &\quad \left. \frac{p_j}{(\rho_\Pi)_j^2} (\ (\nabla \rho_\Pi)_j \cdot \mathbf{v}_j - \nabla \Pi (\rho_\Pi \mathbf{v})_j \ ) \right) = 0 \end{aligned} \tag{1.52}$$

Obviously, along the path of any particle, entropy does not change at all, provided that the time integration is exact. This coincides with the fact that for Euler flow the entropy does not change on the path of a fluid particle provided the fluid particle does not run through a shock. An increase of the entropy production along a particle's path, especially for occurring shocks, is usually invoked by adding an artificial viscosity term. This term is most commonly added as a viscous pressure term p_v to the physical pressure term p. By this method, the rewritten equations for momentum and energy appear as

$$\frac{d\mathbf{v}_j}{dt} = -\nabla \Pi \left(\frac{p + p_v}{\rho_\Pi} \right)_j - \frac{p_j + p_{v,j}}{(\rho_\Pi)_j^2} (\nabla \rho_\Pi)_j \tag{1.53}$$

$$\frac{du_j}{dt} = \frac{p_j + p_{v,j}}{(\rho_\Pi)_j^2} (\ (\nabla \rho_\Pi)_j \cdot \mathbf{v}_j - \nabla \Pi (\rho_\Pi \mathbf{v})_j \ ) \tag{1.54}$$

It is easy to show that entropy now changes dependent on p_v through

$$\frac{ds}{dt} = \frac{1}{T} \cdot \frac{p_v}{(\rho_\Pi)^2} (\ \nabla \rho_\Pi \ \cdot \mathbf{v} - \nabla \Pi (\rho \mathbf{v}) \ ) \tag{1.55}$$

The artificial viscosity was actually involved in order to keep the computations stable for occurring shocks, however the simple calculations above show, that the change of entropy is consistent, if p_v is chosen to be negative for compression ($\nabla \Pi \mathbf{v} < 0$) and zero otherwise.

1.2.5. Complete Set of Equations for Time Evolution in Classical SPH

Summarizing the previous sections, we give the time update of the relevant variables in SPH.

Suppose, at a certain time t_0 there is given a complete particle system for solving the Euler equations

$$\omega_N^c(t_0) = \{\ (m_i, \mathbf{x}_i, \mathbf{v}_i, u_i);\ i = 1...N;\ \mathbf{x}_i \in \Omega(t_0)\ \} \tag{1.56}$$

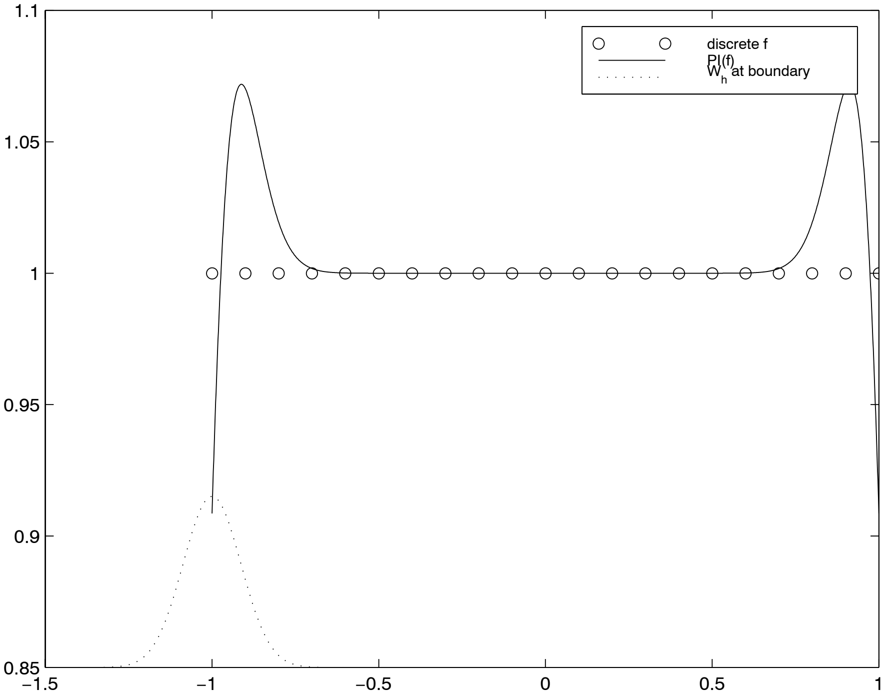
Then we can compute the particle system $\omega_N^c(t)$ at the later time t by the laws of time change of particle information

$$\begin{aligned}
\frac{d}{dt} m_j &= 0 \\
\frac{d}{dt} \mathbf{x}_j &= \mathbf{v}_j \\
\frac{d}{dt} \mathbf{v}_j &= -\nabla \Pi \left(\frac{p}{\rho_\Pi} \right)_j - \frac{p_j}{(\rho_\Pi)_j^2}\ \nabla (\rho_\Pi)_j \\
\frac{d}{dt} u_j &= \frac{p_j}{(\rho_\Pi)_j^2} (\ (\nabla \rho_\Pi)_j \cdot \mathbf{v}_j - \nabla \cdot \Pi (\rho_\Pi \mathbf{v})_j\ )
\end{aligned} \tag{1.57}$$

Note, that the pressure appearing in the equation for conservation of momentum and energy may always be computed in a thermo-dynamical sense as a function of density and internal energy, $p_i = p((\rho_\Pi)_i, u_i)$.

1.2.6. Setting up Boundary Conditions in Classical SPH

While the classical SPH gives reasonable computation results inside of $\Omega(t)$, the method (as introduced in the previous sections) breaks sometimes down at the boundary $\partial\Omega(t)$. The reason is obvious. The approximation quality of the particle approximation Πf of any function f at the boundary is not convenient. As an example, we want to consider the particle approximation of a constant function $f(\mathbf{x}) := 1$ (in 1D for simplicity) at the boundary.



In the plot above, $\Omega = [-1,1]$, and so $\partial\Omega = \{-1;1\}$, $f := 1$. We see that Πf (solid line) reproduces f very well inside of Ω. However, if the distance to the boundary is less than h (in this special case $h = 0.3$), then Πf starts to loose approximation quality. This is due to the fact that W_h is well defined 'behind' the boundary (since W_h has to be symmetric and thus W_h cannot stop right at $\partial\Omega$) but f is not defined outside of Ω. So, W_h expects information of f outside of Ω but there is non. The result of this is seen in the plot above. Classical SPH is, in the introduced form, a non-practical method for treating flows with boundaries. It lacks some additional tricks. The two most important of them are briefly described below.

Boundary Potentials

Monaghan, Morris (see [17],[18],[19]) suggested the idea to use boundary potentials in order neutralize the errors made at the boundary. In this sense, boundary particles are placed at the boundary. The boundary particles induce forces acting on the fluid particles. Lennard-Jones-like potentials are chosen (which, in physics, have the meaning of traction forces between molecules) which are adapted to the SPH setup. A commonly used potential is

$$V(\mathbf{r}) = \frac{\mathbf{r}}{||\mathbf{r}||} \left\{ \begin{array}{cc} D\left(\left(\frac{r_0}{||\mathbf{r}||}\right)^{p_1} - \left(\frac{r_0}{||\mathbf{r}||}\right)^{p_2} \right) & 0 < ||\mathbf{r}|| \leq r_0 \\ 0 & ||\mathbf{r}|| > r_0 \end{array} \right\} \tag{1.58}$$

where $\mathbf{r}$ is the distance between fluid particle and boundary particle, r_0 is the maximum distance of interaction between a fluid and a boundary particle. The order of magnitude of r_0 is typically the same as the one of h. For the exponents one commonly uses $p_1 = 4$ and $p_2 = 2$. D is adapted to the physical problem.

The forces acting from the boundary particles on the fluid particles are introduced in the momentum equation which then appears as

$$\frac{d\mathbf{v}_j}{dt} = -\sum_{i=1}^{N} \left(\frac{p_i}{\Pi\rho_i^2} + \frac{p_j}{\Pi\rho_j^2} \right) \cdot \nabla W_{h,ji} \cdot m_i \quad + \quad \sum_{i=1}^{M} f(\mathbf{x}_j - \bar{\mathbf{x}}_i) \tag{1.59}$$

where M is the number of boundary particles and $\bar{\mathbf{x}}_i$ determines the position of the boundary particle with index i. The schemes for conservation of mass and energy remain untouched.

Treating boundaries in this way means nothing else than bouncing back particles which are too close to the boundary. Thereby one avoids actually the evaluation of approximative derivatives of physical data like density or pressure. Since the introduction of boundary potentials brings no improvement in computing particle approximations in the neighborhood of boundaries, it works well for solid boundaries, but it implies certainly problems when treating inflow or outflow boundaries.

Another attempt in the sense of boundary potentials was made by Hartig (see [13]). He realized that boundary potentials were in some sense uncomfortable since they exist completely independent from the state of flow (density, velocity, and pressure) in the neighborhood of the boundary. Hartig tried to establish a boundary-potential-like formulation which really depends on the state of flow. In order to do this, he tried to give particle approximations of those surface integrals, which arise in the weak formulation of conservation laws. Hartig employs the usual form of a conservation law

$$\frac{\partial \Phi}{\partial t} + \nabla(\Phi \mathbf{v}) + \nabla \mathbf{F} = S \tag{1.60}$$

multiplies it through by an arbitrary test function $g \in C^2(\Omega)$ and writes it down in weak formulation

$$\int_\Omega \frac{\partial \Phi}{\partial t} g + \int_{\partial\Omega} \Phi g(\mathbf{v}^T \cdot \mathbf{n}) - \int_\Omega \Phi \mathbf{v}^T \cdot \nabla g + \int_{\partial\Omega} g(\mathbf{F}^T \cdot \mathbf{n}) - \int_\Omega \mathbf{F}^T \cdot \nabla g = \int_\Omega Sg \tag{1.61}$$

where Φ is the conservation variable and $\mathbf{F}$ is the flux. By assuming that $\mathbf{v}^T \cdot \mathbf{n} = 0$ at the boundary (true for solid walls) and after reordering one obtains

$$\int_\Omega \frac{\partial \Phi}{\partial t} g - \int_\Omega \Phi \mathbf{v}^T \cdot \nabla g - \int_\Omega \mathbf{F}^T \cdot \nabla g - \int_\Omega Sg + \int_{\partial\Omega} g(\mathbf{F}^T \cdot \mathbf{n}) = 0 \tag{1.62}$$

In the case of the Euler equations, the particle approximation of the first four volume integrals and the clever choice of g leads to a system of equations similar to equations (1.34) and (1.41). Thus, it is left to give a suitable approximation of the remaining surface integral. In this sense, Hartig comes up with

$$\frac{d}{dt}(\alpha_j \Phi_j) = -\sum_{i=1}^{N} \alpha_j \alpha_i (\ \mathbf{F}_i + \mathbf{F}_j\ ) \cdot \nabla W_{h,ji} \ + \ 2\alpha_j \mathbf{F}_j^T \cdot \mathbf{OI}_j \ + \ \alpha_j S_j \tag{1.63}$$

where

$$\mathbf{OI}_j = \int_{\partial\Omega} W_h(\mathbf{x} - \mathbf{x}_i)\mathbf{n}$$

and the α_j are the weights of the particles. Hartig tries to give approximate solutions to $\mathbf{OI}_j$ and demonstrates good results of this method, for example reflection of shocks on solid walls. Obviously, this method works well. Unfortunately, Hartig did not extend this work to the case of inflow, outflow, or free surface boundaries. Then, the evaluation of another surface integral comes into play. About this, there is no research done, yet.

Modified Smoothing Kernels

Another method is practiced by Vila (see [31]). He suggests to introduce modified smoothing kernels. Obviously, when considering particles close to the boundary, symmetric smoothing kernels reach over the boundary and hence

$$\int_\Omega W_h(\mathbf{y} - \mathbf{x}) d\mathbf{x} \neq 1 \tag{1.64}$$

if $\mathbf{y}$ is close enough to the boundary. This, by the way, is the reason for all wrong smoothed particle approximation results near $\partial\Omega$. There are two ways to get around this unfortunate circumstances.

- One attempt is to re-normalize the smoothing kernel such that

$$\int_{\Omega} W_h(\mathbf{y}-\mathbf{x})d\mathbf{x} \overset{!}{=} 1 \tag{1.65}$$

  is satisfied.

- It is possible to introduce ghost particles, which are particles in Ω reflected by the boundary into $\mathbf{R}^{\nu} \setminus \Omega$ (thus penetrating information from 'inside Ω' to 'outside Ω' and therewith considerably improving approximation properties)

In those two ways, no additional terms need to be added to the equations of time evolution. Renormalized kernels proof to provide good results in time evolution but they do not prove to provide discrete conservation of energy and momentum. Ghost particles do provide discrete conservation of momentum but they are difficult to handle boundaries having corners, sharp edges, or bendings.

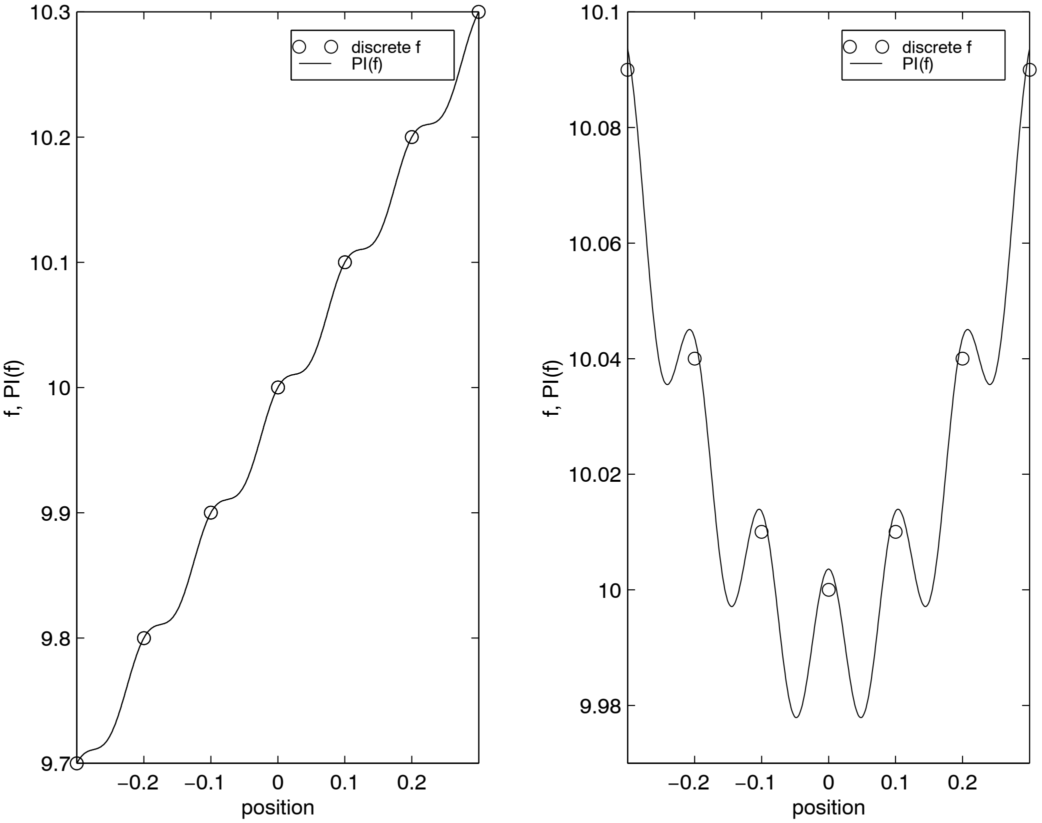
1.2.7. Sketch of Some Unpleasant Properties of Classical SPH

We would like to give a summary of unpleasant properties of the classical SPH method.

- As wee have already seen in section 1.2.6, the Π-approximation looses quality at the boundary.

- The quality of approximation always depends on the distance $D(\delta_{\omega_N}, \mu_{\rho_\Pi})$ between the two measures. Although D can be small at a certain time t_0, it might become large because of particles becoming disordered

- We would like to outline some behavior of Πf if h becomes very small compared to the particle's distances. In the plots below we exhibit the approximations of a linear and of a quadratic function in 1D, here with $h = 0.175$.



1.3. Conclusions

The previous sections showed that SPH provides a very nice formulation for approximation of the in-stationary, compressible Euler equations. Of certain advantage is that

- classical SPH can be quickly implemented and well be parallelized with no much effort
- classical SPH is a grid free method, no mesh needs to be established or reorganized i.e. moving geometry is easy to handle in the computation)

- classical SPH is a very fast method, since computing Πf requires simple operations.

Of certain disadvantage is that

- problems occur when we attempt to treat boundary conditions
- the local approximation quality is uncertain even inside of Ω
- the second derivative of a smooth function Πf is not employable (see right plot above where the discrete function values describe a function of positive second derivatives, but the smoothed function describes a negative second derivative at the discrete positions)
- the classical SPH does not provide the necessary entropy production when particles run through shocks.

The aim of this thesis is to keep all the good properties and find a practicable solution to all the disadvantageous items.

In chapter 2 we will introduce a new particle model which is based on general conservation laws. This model is close to the classical SPH model, but has some properties which provide some advantages.

In chapter 3 we are going to establish a general ansatz for a smoothed particle approximation Πf which provides much better approximation properties than the one which was introduced in chapter 1.

In chapter 4 finally we will speak about treatment of boundary conditions using the particle model introduced in chapter 2.

Chapter 5 provides us with some useful technical items, i.e. sorting and searching for nearest neighbor particles as well as adding and removing particles.

Chapter 6 finally presents an upwind method for the particle scheme, introduced in chapter 2.

Chapter 2

2. General SPH - A New Particle Scheme for Conservation Laws

One of the conclusions of chapter 1 was that the classical SPH setup lacks good approximation properties at the boundary in general, and there are cases where approximation properties might become very bad anywhere in $\Omega(t)$. This is due to the employment of symmetric kernels. Hence, the classical SPH approach is difficult to handle for boundary treatment.

From now on we would like to consider general smoothed particle approximations Πf which are not necessarily based on symmetric smoothing kernels but which will provide much better approximation properties inside of the flow domain and especially at the boundary. Avoiding symmetric kernels means that discrete approximation properties will not hold any more. We will therefore have to come up with a new numerical particle scheme that works with a so called general smoothed particle approximation and that still provides some convenient conservation properties as well.

The following definition explains, what we understand by a general smoothed particle approximation of an arbitrary function f.

Definition 2.1 (General Smoothed Particle Operator/Approximation)
Let $\mathcal{W}_N(\Omega(t)) := \{\ \omega_N(t) \mid \omega_N(t)$ is a particle ensemble in $\Omega(t),\ t \in T\ \}$ be the set of particle ensembles in $\Omega(t)$. Let $h : \bar{\Omega} \to \mathbf{R}^+$ denote the smoothing length, i.e. the maximum interaction radius between two particles and let $h_i := h(t, \mathbf{x}_i)$ at a fixed time t. We call the operator

$$\Pi^n$$

a **general smoothed particle operator** if it has the following properties

- For a fixed time t we have
  $\Pi^n : \left\{\mathcal{W}_N(\Omega(t)) \times \mathbf{R}^N\right\} \longrightarrow C^n(\Omega(t)) \quad$ for some $n \in \mathbf{N}$
  i.e. the operator Π^n makes a smooth function out of a particle system and an assigned vector of discrete values.

Convention:
Let $\mathbf{f} : T \to \mathbf{R}^N$, $\mathbf{f}(t) := \{\ f_1(t), ..., f_N(t)\ \}$. We define the symbol of the smooth function produced by the operator Π^n as
$\Pi^n_{\omega_N} f(t, \mathbf{y})\ := \Pi^n(\ \omega_N(t), \mathbf{f}(t)\ ),\ \mathbf{y} \in \Omega(t)$.
We would like to emphasize that throughout this thesis, $\mathbf{f}(t)$ is viewed as a vector containing N discrete values known at the N particle positions. In this sense, $\Pi^n_{\omega_N} f$ is a **general smoothed particle approximation** of the discrete value set $\mathbf{f}(t)$, in other words it is a smooth function made out of the discrete particle values.

Convention:
Occasionally within this thesis, we abbreviate
$\Pi f := \Pi^n_{\omega_N} f$
$(\Pi f)_i = (\Pi^n_{\omega_N} f)_i = \Pi f_i = \Pi^n_{\omega_N} f_i := \Pi^n_{\omega_N} f(t, \mathbf{x}_i)$

Convention:
If g is a function $g : \bar{\Omega} \to \mathbf{R}$, then we denote (similarly to what was denoted above)
$\Pi^n_{\omega_N} g(t, \mathbf{y})\ :=\ \Pi^n(\ \omega_N(t), \mathbf{g}(t)\ ),\ \mathbf{y} \in \Omega(t)$, where $\mathbf{g}(t) := \{\ g(t, \mathbf{x}_1), ..., g(t, \mathbf{x}_N)\ \}$

Convention:
Let $\mathbf{v}_i : T \to \mathbf{R}^\alpha;\ \ i = 1, ..., N;\ \ \alpha \in \mathbf{N}$, $\mathbf{v}_i(t) = \{\ v_i^{(j)}(t);\ j = 1, ..., \alpha\ \}$. Then we denote
$\Pi^n_{\omega_N} \mathbf{v} := \left\{\ \Pi^n_{\omega_N} v^{(j)};\ j = 1, ..., \alpha\ \right\}$

Convention:
Let $\mathbf{w} : \bar{\Omega} \to \mathbf{R}^\alpha$ with $\mathbf{w}(t, \mathbf{y}) = (w^{(j)}(t, \mathbf{y});\ j = 1, ..., \alpha)$, $\mathbf{y} \in \Omega(t)$ then we denote
$\Pi^n_{\omega_N} \mathbf{w} := \left\{\ \Pi^n_{\omega_N} w^{(j)}\ ;\ \ j = 1, ..., \alpha\ \right\}$

- $\Pi^n_{\omega_N}(f + g) = \Pi^n_{\omega_N} f + \Pi^n_{\omega_N} g$
  $\Pi^n_{\omega_N}(c \cdot f) = c \cdot \Pi^n_{\omega_N} f \qquad c = \text{const}$

- $\frac{\partial}{\partial \mathbf{x}_i} \Pi^n_{\omega_N} f(t, \mathbf{y})$ is continuous $\forall \mathbf{y} \in \Omega(t)$
  and we have $\frac{\partial}{\partial \mathbf{x}_i} \Pi^n_{\omega_N} f(t, \mathbf{y}) = 0$ if $\|\mathbf{y} - \mathbf{x}_i\|_2 > h_i\ \ \forall i \in \{1, ..., N\}$

- $\frac{\partial}{\partial f_i} \Pi^n_{\omega_N} f(t, \mathbf{y})$ is continuous $\forall \mathbf{y} \in \Omega(t)$
  and we have $\frac{\partial}{\partial f_i} \Pi^n_{\omega_N} f(t, \mathbf{y}) = 0$ if $\|\mathbf{y} - \mathbf{x}_i\|_2 > h_i\ \ \forall i \in \{1, ..., N\}$

where $h_i = h(t, \mathbf{x}_i)$, with $h : \bar{\Omega} \to \mathbf{R}^+$ is a given function and denotes the smoothing length (i.e. the maximum interaction radius of any two particles).
Remark: The classical smoothed particle approximation Πf belongs clearly to the set of general smoothed particle approximations in the sense stated above.

At this point, we do not care about approximation properties, so $\Pi^n_{\omega_N} f$ might be of good or bad approximation quality. Please refer to chapter 3 where we are going to exhibit a whole class of general smoothed particle approximations. However, we give a definition which determines the 'distance' between f and $\Pi^n_{\omega_N} f$.

Definition 2.2:
Given a particle ensemble $\omega_N(t)$. Given $f : \bar{\Omega} \to \mathbf{R}$, and $\Pi^n_{\omega_N} f$ a general smoothed particle approximation of f in the sense of definition 2.1. We call

$$E\Pi^n_{\omega_N} f := \Pi^n_{\omega_N} f - f \tag{2.1}$$

the local approximation error.

2.1. Partial Differential Equation to be Approximated

With the definitions made above we are ready to introduce a general smoothed particle scheme. Before this scheme will be defined (in section 2.2), we first have a brief look at the partial differential equations which we would like to approximate by our numerical scheme. In general, these are conservation laws given in

Definition 2.3: (Conservation Laws)
Let $\Omega(t)$ be the domain of fluid flow. Let $\mathbf{v} : \bar{\Omega} \to \mathbf{R}^\nu$ be the velocity field of the fluid and let t_0 denote the initial point of time. Then the general formulation of a conservation law is given by

$$\frac{d}{dt} \int\limits_{\mathbf{T}_A(t)} \Phi \, d\mathbf{x} \; + \int\limits_{\partial \mathbf{T}_A(t)} \mathbf{F}^T \cdot \mathbf{n} \, d\sigma \; = \int\limits_{\mathbf{T}_A(t)} S \tag{2.2}$$

where

- Φ is the conservation variable

- the set $\mathbf{T}_A(t)$ is defined by

$$\mathbf{T}_A(t) := \{ \, \mathbf{x}(t) \;|\; \mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(\tau, \mathbf{x}(\tau)) \, d\tau, \; \mathbf{x}_0 \in A \, \} \tag{2.3}$$

  and $A \subset \Omega(t_0)$ is an arbitrary subset with regular boundary.

- $\mathbf{F}$ is the flux vector and is usually a function of Φ
- $\mathbf{n}$ is the outward pointing normal of $\partial \mathbf{T}_A(t)$
- S is the source term.

This general form of conservation laws is given by classical physics. However, in this thesis we restrict ourself to the case of viscid or in-viscid fluid flow, which means that we consider only special cases of the general form of conservation laws.

Definition 2.4 (Euler/Navier-Stokes equations)
The equations describing viscid (Navier-Stokes) or in-viscid (Euler) fluid flow are given by the specifications below.

$$\Phi = (\rho, \rho \mathbf{v}^T, \rho E) \tag{2.4}$$

$$\mathbf{F}^{Eu} \;=\; (\;\; \mathbf{0}, \;\; p\,\mathbf{I}, \;\; (p\,\mathbf{I}) \cdot \mathbf{v} \;\;) \qquad \text{(Euler)} \tag{2.5}$$

$$\mathbf{F}^{NS} \;=\; (\;\; \mathbf{0}, \;\; p\,\mathbf{I} - \eta\,\mathbf{St}, \;\; (p\,\mathbf{I} - \eta\,\mathbf{St}) \cdot \mathbf{v} - \lambda\,\nabla T \;\;) \qquad \text{(Navier-Stokes)} \tag{2.6}$$

where $\mathbf{St}$ denotes the deformation tensor and is represented by

$$\mathbf{St}^{(i,j)} = \frac{\partial v^{(i)}}{\partial x^{(j)}} + \frac{\partial v^{(j)}}{\partial x^{(i)}} - \frac{2}{3} \nabla \mathbf{v} \delta_{ij} \tag{2.7}$$

and the sources are given by

$$S = (\, 0, \mathbf{g}^T, \mathbf{v}^T \cdot \mathbf{g} \,) \tag{2.8}$$

We have

- $\mathbf{g}$ = the gravity vector
- $\mathbf{v} = (v^{(1)}, v^{(2)}, v^{(3)})^T$, as usual, the velocity vector

- $E = u + \frac{1}{2}\mathbf{v}^T \cdot \mathbf{v}$ the total energy of the fluid.
- u the internal energy defined by $du = dq + \frac{p}{\rho^2} d\rho$ (q = heat, p=pressure, ρ=density).
- η is the viscosity of the fluid
- λ the coefficient of heat conduction of the fluid.

Now, we come to the very important introduction of the numerical scheme.

2.2. General Smoothed Particle Scheme for Conservation Laws

The conservation law in the form (2.2) is true for any measurable subset $A \subset \Omega(t_0)$ with regular boundary. Hence, for this reason, equation (2.2) is usually rewritten as

$$\frac{d\Phi}{dt} + (\nabla \mathbf{v}) \cdot \Phi + \nabla \mathbf{F} = S \tag{2.9}$$

Equation (2.9) is equivalent to (2.2) for all cases, where **F** and **v** are smooth. However, if **F** and **v** prove to have discontinuities, equation (2.9) must not be employed and one would have to step back to the integral form (2.2).

In the following, we will have to introduce a numerical approximation scheme based either on equation (2.2) or on equation (2.9) since both are equivalent in the smooth case. We choose equation (2.9) since here we have the chance to give approximations of the appearing derivatives of the functions **F** and **v** by general smoothed particle approximations. Thus, the numerical scheme to be introduced is, like in classical SPH, based on the fact, that a complete particle ensemble $\omega_N^c(t)$ is established in $\Omega(t)$ with particles being carriers of information. Consequently, all appearing functions Φ, $\mathbf{v}$, $\mathbf{F}$ and S will usually be known only at the discrete particle locations. The whole work consists in giving approximations of the appearing derivatives by a general smoothed particle approximation.

At this point, we give the definition of the general smoothed particle scheme. After this, we turn to exhibiting some important properties of this scheme concerning conservation and time integration.

Definition 2.5: (General smoothed particle scheme for conservation laws) Given a flow domain $\Omega(t)$. Let $\omega_N^c(t) := \{(\ \mathbf{x}_i(t), m_i, \Phi_i(t)\ );\ i = 1...N\}$ be a complete particle system established in $\Omega(t)$, with Φ the vector of conservation variables completely describing the physical problem. Let $\overset{\circ}{\Phi}$: $\Omega(t_0) \rightarrow \mathbf{R}^\alpha$ be the initial condition at the time t_0 to the physical problem to be solved and assume it is smooth; let $\overset{\circ}{\mathbf{x}}_i$ be the initial position of particle i. We call the numerical approximation scheme

$$\begin{aligned}\frac{d}{dt}\Phi_i &:= -(\nabla\Pi^n_{\omega_N}\mathbf{v}_i)\,\Phi_i \;-\; \nabla\Pi^n_{\omega_N}\mathbf{F}_i + S_i && (2.10)\\ \frac{d}{dt}\mathbf{x}_i &:= \Pi^n_{\omega_N}\mathbf{v}_i && (2.11)\end{aligned}$$

with

$$\Pi^n_{\omega_N}\mathbf{v}_i := \Pi^n_{\omega_N}\mathbf{v}(t,\mathbf{x}_i) \qquad \Pi^n_{\omega_N}\mathbf{F}_i := \Pi^n_{\omega_N}\mathbf{F}(t,\mathbf{x}_i) \qquad S_i = S(t,\mathbf{x}_i)$$

a general smoothed particle scheme for conservation laws of the type (2.2). The initial condition to this set of ODE is given by

$$\begin{aligned}\Phi_i(t_0) &:= \overset{\circ}{\Phi}\,(\overset{\circ}{\mathbf{x}}_i) \;=\overset{\circ}{\Phi}_i && (2.12)\\ \mathbf{x}_i(t_0) &:= \overset{\circ}{\mathbf{x}}_i && (2.13)\end{aligned}$$

With this, we define the set of solutions $\mathbf{L}^n(t)$ simply by

$$\mathbf{L}^n(t) := \{\;(\Phi_i(t),\mathbf{x}_i(t)) \mid i \in \{1;...;N\}\;\} \tag{2.14}$$

In an equivalent, but more general way we rewrite system (2.10/2.11) as

$$\frac{d\mathbf{y}_i}{dt} \;=\; \mathbf{g}(\mathbf{y}_i) \tag{2.15}$$

where

$$\mathbf{y}_i = (\rho_i, \rho_i v_i^{(1)}, \rho_i v_i^{(2)}, \rho_i v_i^{(3)}, \rho_i E_i, x_i^{(1)}, x_i^{(2)}, x_i^{(3)}) \tag{2.16}$$

and

$$\mathbf{g}(\mathbf{y}_i) = \begin{pmatrix} -(\nabla\Pi^n_{\omega_N}\mathbf{v}_i)\Phi_i \;-\; \nabla\Pi^n_{\omega_N}\mathbf{F}_i + S_i \\ \Pi^n_{\omega_N}\mathbf{v}_i \end{pmatrix} \tag{2.17}$$

In the most general way we may write

$$\frac{d\mathbf{Y}}{dt} \;=\; \mathbf{G}(\mathbf{Y}) \tag{2.18}$$

where

$$\mathbf{Y} := (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N) \tag{2.19}$$

and

$$\mathbf{G} := (\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_N) \tag{2.20}$$

We realize that establishing the new, general smoothed particle scheme leads, as in classical SPH, to a system of ordinary differential equations.

After having established the new particle scheme, we now have to think about what conservation properties do characterize the new setup (which will be done in detail in section 2.3). Moreover, we need to think about how to numerically integrate the set of ordinary differential equations (which will be shown in section 2.4).

2.3. Conservation properties

2.3.1. Conservation of Mass, Momentum and Energy

The scheme stated in (2.10/2.11) is obviously the direct approximation of equation (2.9). Since it was developed to numerically compute conservation laws, we have to show that this scheme indeed provides certain conservation properties. We would like to do this by showing that our scheme does not solve exactly the original conservation law, but it solves directly some similar law which is, in the physical sense, also a conservation law.

The scheme (2.10/2.11) is defined for the time evolution of the particles only, through which some discrete set of solutions $\mathbf{L}^n(t)$ is defined. Unlike for classical SPH, we will not be able to show some kind of discrete conservation properties concerning $\mathbf{L}^n(t)$. However, we will show continuous conservation properties. For this purpose we write the system (2.10/2.11) in a more global way, namely we define the time evolution not only for the particles, but also for all other points $\mathbf{x}_0 \in \Omega(t_0)$ which do not coincide with some particle positions. We can easily do this since the general smooth particle approximations are defined throughout $\Omega(t)$, not only at the particle positions.

Definition 2.6
Given a flow domain $\Omega(t)$. Let $\omega_N^c(t) := \{(\ \mathbf{x}_i(t), m_i, \Phi_i(t)\ )\,; i = 1...N\}$ be a complete particle system established in $\Omega(t)$ being the same as given in definition 2.5. Let $\overset{\circ}{\Phi}$: $\Omega(t_0) \to \mathbf{R}^\alpha$ be the initial condition at the time t_0 and assume it is smooth; let $\overset{\circ}{\mathbf{x}}_i$ be the initial position of particle i at time t_0. Let Π^n be the same general smoothed particle operator as used in definition 2.5. We define the global

smoothed particle scheme by

$$\frac{d}{dt}\Phi_{(\mathbf{x}_0)} := -\Phi_{(\mathbf{x}_0)}\,\nabla\Pi^n_{\omega_N}\mathbf{v}(t,\mathbf{x}_{(\mathbf{x}_0)}) \;-\; \nabla\Pi^n_{\omega_N}\mathbf{F}(t,\mathbf{x}_{(\mathbf{x}_0)}) + S(t,\mathbf{x}_{(\mathbf{x}_0)}) \quad (2.21)$$

$$\frac{d}{dt}\mathbf{x}_{(\mathbf{x}_0)} := \Pi^n_{\omega_N}\mathbf{v}(t,\mathbf{x}_{(\mathbf{x}_0)}) \quad (2.22)$$

$\mathbf{x}_{(\mathbf{x}_0)}$ is the position of the virtual fluid particle that was initially at the position $\mathbf{x}_0$. Therefore, $\Phi_{(\mathbf{x}_0)}$ is the vector of conservation variables of the virtual particle that was initially at position $\mathbf{x}_0$.

The initial conditions are

$$\Phi_{(\mathbf{x}_0)}(t_0) = \overset{\circ}{\Phi}(\mathbf{x}_0) \quad \forall\mathbf{x}_0 \in \Omega(t_0) \quad (2.23)$$

$$\mathbf{x}_{(\mathbf{x}_0)}(t_0) = \mathbf{x}_0 \quad \forall\mathbf{x}_0 \in \Omega(t_0) \quad (2.24)$$

We define the set of solutions to the above stated set of ODE's by

$$\mathcal{L}^n(t) := \{\,(\Phi_{(\mathbf{x}_0)}(t),\mathbf{x}_{(\mathbf{x}_0)}(t)) \mid \mathbf{x}_0 \in \Omega(t_0)\} \quad (2.25)$$

For any particle with index i, we find the time evolution of its quantities in $\mathcal{L}^n(t)$, because $\Phi_i(t) = \Phi_{\left(\overset{\circ}{\mathbf{x}}_i\right)}(t)$ and $\mathbf{x}_i(t) = \mathbf{x}_{\left(\overset{\circ}{\mathbf{x}}_i\right)}(t)$. Therefore, we clearly have

$$\mathbf{L}^n(t) \subset \mathcal{L}^n(t) \quad (2.26)$$

All our further considerations about conservation properties will be made concerning the set of ODE (2.21/2.22) knowing that this set is in some sense the globalization of the particle scheme (2.10/2.11), i.e. knowing that, with $\mathbf{L}^n(t)$, we just compute a subset of the complete solution set $\mathcal{L}^n(t)$.

Theorem 2.7: (Conservation properties of the Global Smoothed Particle Scheme)
The system of ODE (2.21/2.22) represents a conservation law.

Proof:
Since we consider all points $\mathbf{x}_0 \in \Omega(t_0)$, we can omit the index '$\mathbf{x}_0$' in the system of equations and may rewrite (2.21/2.22) as

$$\frac{d}{dt}\Phi = -(\nabla\Pi^n_{\omega_N}\mathbf{v})\Phi \;-\; \nabla\Pi^n_{\omega_N}\mathbf{F} + S \quad (2.27)$$

$$\frac{d}{dt}\mathbf{x} = \Pi^n_{\omega_N}\mathbf{v} \quad (2.28)$$

This amounts to consider all $\mathbf{x} \in \Omega(t)$. At any time t, we may take the integral of equation (2.27) over an arbitrary region $\Pi\mathbf{T}_A(t)$ with smooth boundary which is defined by

$$\Pi\mathbf{T}_A(t) := \{\ \mathbf{x}(t)\ \mid\ \mathbf{x}(t) = \mathbf{x}_0 + \int\limits_{t_0}^{t} \Pi^n_{\omega_N}\mathbf{v}(\tau, \mathbf{x}(\tau))\, d\tau,\ \mathbf{x}_0 \in A, A \subset \Omega(t_0)\ \} \tag{2.29}$$

Thus we can write

$$\int\limits_{\Pi\mathbf{T}_A(t)} \frac{d}{dt}\Phi d\omega\ +\ \int\limits_{\Pi\mathbf{T}_A(t)} (\nabla\Pi^n_{\omega_N}\mathbf{v})\Phi d\omega\ +\ \int\limits_{\Pi\mathbf{T}_A(t)} \nabla\Pi^n_{\omega_N}\mathbf{F} d\omega\ =\ \int\limits_{\Pi\mathbf{T}_A(t)} S\, d\omega \tag{2.30}$$

By the smoothness of $\Pi^n_{\omega_N}\mathbf{v}$ and Φ we rewrite (2.30) as

$$\begin{aligned} \int\limits_{\Pi\mathbf{T}_A(t)} \left(\frac{\partial\Phi}{\partial t} + \nabla\Phi\cdot\Pi\mathbf{v}\right) d\omega\ +\ \int\limits_{\Pi\mathbf{T}_A(t)} (\nabla\Pi^n_{\omega_N}\mathbf{v})\Phi d\omega\ +& \\ \int\limits_{\Pi\mathbf{T}_A(t)} \nabla\Pi^n_{\omega_N}\mathbf{F} d\omega\ =\ \int\limits_{\Pi\mathbf{T}_A(t)} S\, d\omega& \end{aligned} \tag{2.31}$$

which is equivalently

$$\int\limits_{\Pi\mathbf{T}_A(t)} \frac{\partial\Phi}{\partial t} d\omega\ +\ \int\limits_{\Pi\mathbf{T}_A(t)} \nabla(\Pi^n_{\omega_N}\mathbf{v}\cdot\Phi) d\omega\ +\ \int\limits_{\Pi\mathbf{T}_A(t)} \nabla\Pi^n_{\omega_N}\mathbf{F} d\omega\ =\ \int\limits_{\Pi\mathbf{T}_A(t)} S d\omega \tag{2.32}$$

Now we rewrite the second and third term in this sum as surface integrals, yielding

$$\int\limits_{\Pi\mathbf{T}_A(t)} \frac{\partial\Phi}{\partial t} d\omega\ +\ \int\limits_{\partial\Pi\mathbf{T}_A(t)} (\Pi^n_{\omega_N}\mathbf{v}\cdot\Phi)^T\, \mathbf{n} d\sigma\ +\ \int\limits_{\partial\Pi\mathbf{T}_A(t)} (\Pi^n_{\omega_N}\mathbf{F})^T \mathbf{n}\, d\sigma\ =\ \int\limits_{\Pi\mathbf{T}_A(t)} S d\omega \tag{2.33}$$

which finally provides

$$\frac{d}{dt}\int\limits_{\Pi\mathbf{T}_A(t)} \Phi d\omega\ +\ \int\limits_{\partial\Pi\mathbf{T}_A(t)} (\Pi^n_{\omega_N}\mathbf{F})^T \mathbf{n}\, d\sigma\ =\ \int\limits_{\Pi\mathbf{T}_A(t)} S d\omega \tag{2.34}$$

Equation (2.34) is clearly in the general form of conservation laws as exhibited in equation (2.2) and so the proof is complete.

Equation (2.34) differs form the exact conservation law in fluid mechanics only in two things:

- The flux over the boundary of the control volume $\Pi\mathbf{T}_A(t)$ is $\Pi^n_{\omega_N}\mathbf{F}$ instead of $\mathbf{F}$.
- The control volume $\Pi\mathbf{T}_A(t)$ is moved with $\Pi^n_{\omega_N}\mathbf{v}$ instead of $\mathbf{v}$.

With the general smoothed particle scheme we solve exactly for solutions of a law of conservation type which is not the original conservation law, but it is nothing else than the particle approximation of the original conservation law. This leads to a very interesting conclusion. With any general smoothed particle approximation operator Π^n we are able to compute smooth approximations not only of functions, but also of conservation laws.

2.3.2. Conservation of the Global Error

Equation (2.34) indicates that the numerical solution comes the closer to the exact solution the better the approximation quality of $\Pi^n_{\omega_N}\mathbf{v}$ and $\Pi^n_{\omega_N}\mathbf{F}$. In the next theorem we would like to have a look at the errors actually made by the smoothed particle approximation of conservation laws.

Theorem 2.8: (Error estimation for numerical scheme)
Let Φ be the solution provided by the global smoothed particle approximation of conservation laws (equations 2.21/2.22). Let $\bar{\Phi}$ be the solution to the conservation law (2.2), rewritten as $\frac{d_{\mathbf{v}}\bar{\Phi}}{dt} = -\bar{\Phi}\,\nabla\bar{\mathbf{v}} \;-\; \nabla\bar{\mathbf{F}} \;+\; S$ and suppose, $\bar{\Phi}$ is smooth. The error induced by the global smoothed particle approximation of conservation laws (2.21/2.22) is given by

$$\frac{d_{\Pi\mathbf{v}}\, D\Phi}{dt} = -D\Phi \cdot \nabla\Pi^n_{\omega_N}\mathbf{v} - \nabla\left(\, D\mathbf{v}\cdot\bar{\Phi} + D\mathbf{F}\,\right) \tag{2.35}$$

where $D\Phi := \Phi - \bar{\Phi}$, $D\mathbf{v} := \Pi^n_{\omega_N}\mathbf{v} - \bar{\mathbf{v}}$, $D\mathbf{F} := \Pi^n_{\omega_N}\mathbf{F} - \bar{\mathbf{F}}$ and the operator $\frac{d_{\Pi\mathbf{v}}}{dt} := \frac{\partial}{\partial t} + \Pi^n_{\omega_N}\mathbf{v}^T\cdot\nabla$

Proof:
We rewrite the conservation law to be solved by the numerical scheme (2.21/2.22) in differential form:

$$\frac{d_{\Pi\mathbf{v}}\Phi}{dt} + \Phi\cdot\nabla\Pi^n_{\omega_N}\mathbf{v} = -\nabla\Pi^n_{\omega_N}\mathbf{F} + S \tag{2.36}$$

The exact conservation law in differential form producing $\bar{\Phi}$ is given by

$$\frac{d_{\mathbf{v}}\bar{\Phi}}{dt} + \bar{\Phi}\cdot\nabla\bar{\mathbf{v}} = -\nabla\bar{\mathbf{F}} + S \tag{2.37}$$

where the total differential operators are defined by

$$\frac{d_{\Pi \mathbf{v}}}{dt} := \frac{\partial}{\partial t} + \Pi^n_{\omega_N} \mathbf{v}^T \cdot \nabla^T$$

$$\frac{d_{\mathbf{v}}}{dt} := \frac{\partial}{\partial t} + \bar{\mathbf{v}}^T \cdot \nabla^T$$

We can rewrite (2.37) by

$$\frac{d_{\Pi \mathbf{v}} \bar{\Phi}}{dt} + \bar{\Phi} \cdot \nabla \bar{\mathbf{v}} - \nabla \bar{\Phi} \cdot (\Pi^n_{\omega_N} \mathbf{v} - \bar{\mathbf{v}}) = -\nabla \bar{\mathbf{F}} + S \tag{2.38}$$

which furthermore gives

$$\frac{d_{\Pi \mathbf{v}} \bar{\Phi}}{dt} + \bar{\Phi} \cdot \nabla \Pi^n_{\omega_N} \mathbf{v} - \nabla (\ \bar{\Phi} \cdot (\Pi^n_{\omega_N} \mathbf{v} - \bar{\mathbf{v}})\ ) = -\nabla \bar{\mathbf{F}} + S \tag{2.39}$$

We subtract (2.39) from (2.36) and obtain

$$\frac{d_{\Pi \mathbf{v}} (\Phi - \bar{\Phi})}{dt} + (\Phi - \bar{\Phi}) \cdot \nabla \Pi^n_{\omega_N} \mathbf{v} + \nabla (\ \bar{\Phi} \cdot (\Pi^n_{\omega_N} \mathbf{v} - \bar{\mathbf{v}})\ ) = -\nabla (\Pi^n_{\omega_N} \mathbf{F} - \bar{\mathbf{F}}) \tag{2.40}$$

which finally yields

$$\frac{d_{\Pi \mathbf{v}} D\Phi}{dt} = -D\Phi \cdot \nabla \Pi^n_{\omega_N} \mathbf{v} - \nabla \left(\ D\mathbf{v} \cdot \bar{\Phi} + D\mathbf{F}\ \right) \tag{2.41}$$

This concludes the proof.
Remark: When we define

$$D\mathcal{F} := D\mathbf{v} \cdot \bar{\Phi} + D\mathbf{F}$$

then we realize, that the time evolution of the error provided by the numerical approximation scheme is again in conservation form, given by

$$\frac{d\, D\Phi}{dt} + D\Phi \cdot \nabla \Pi^n_{\omega_N} \mathbf{v} + \nabla\, D\mathcal{F} = 0$$

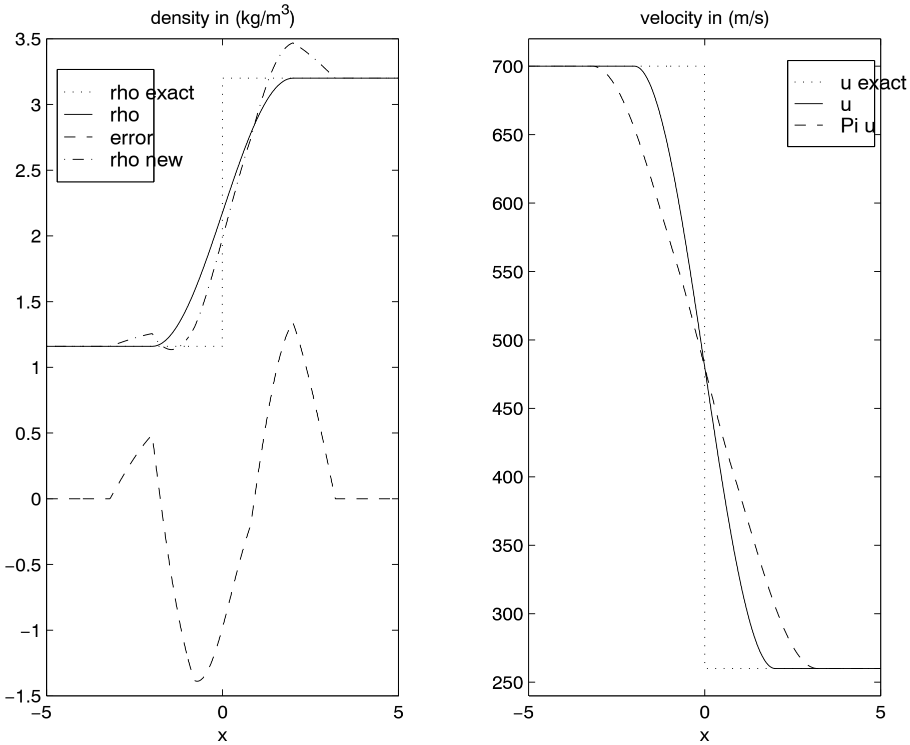
With the help of this error approximation it would be very simple to establish a numerical scheme that would compute numerical solutions more accurately than the scheme (2.10/2.11). The numerical scheme we have in mind is the following.
1. For some time t compute the solutions Φ_i, $i = 1, ..., N$.
2. Using some approximation of $\nabla\, D\mathcal{F}$, we compute an approximation of $D\Phi_i$, $i = 1, ..., N$.
3. Using the approximation of $D\Phi_i$, we compute an approximation Φ_i^c, $i = 1, ..., N$ in the sense of correcting the numerical solution by $\Phi_i^c = \Phi_i - D\Phi_i$ (the little 'c'

stands for 'corrected'), assuming that the approximation derived in this way is closer to $\bar{\Phi}_i$ than Φ_i.

No research was done on this topic yet, but we consider it being worth while because this would provide a means of high accuracy smoothed particle computations.

2.3.3. Source of Instabilities Due to the General SPH Scheme

The error tracking shown in the last section is also useful for explaining some sources of instabilities of the numerical scheme for the case when shocks have to be computed. As a simple example, we would like to consider a 1D shock. In the plot below, we have sketched the density and the velocity of the shock. The example here is a stationary shock. The dotted lines in the plots represent the exact solutions $\bar{\rho}$ and $\bar{u}$. Now assume, at time t we have also some numerical solution ρ and u to this problem, marked by the solid lines in the plot.



Theorem 2.8 states that

$$\frac{d(D\rho)}{dt} = -(D\rho)\,\frac{\partial}{\partial x}(\Pi u) \;-\; \frac{\partial}{\partial x}((\Pi u - \bar{u})\,\bar{\rho})$$

which is, since $\bar{\rho}\,\bar{u} = const$ for stationary 1D shocks,

$$\frac{d(D\rho)}{dt} = -(D\rho)\,\frac{\partial}{\partial x}(\Pi u) \;-\; \frac{\partial}{\partial x}(\Pi u\,\bar{\rho})$$

For an analytical treatment, it is now convenient to consider the partial time derivative of $D\rho$ instead of the total time derivative. This would help us to see, if the computed solutions are stationary, too. So, with $\frac{d(D\rho)}{dt} = \frac{\partial(D\rho)}{\partial t} + \Pi u\,\frac{\partial D\rho}{\partial x}$ we obtain

$$\frac{\partial(D\rho)}{\partial t} = -\frac{\partial}{\partial x}(\rho\,\Pi u)$$

In the left plot above, the dashed line sketches the term $\frac{\partial(D\rho)}{\partial t}$ for this special case. At the down-stream side from the shock front (right from the shock front), we observe that $\frac{\partial(D\rho)}{\partial t}$ is mostly positive, especially at locations where $D\rho = 0$. Consequently, at those positions, $D\rho$ is going to be positive, hence the numerical solution ρ is going to rise above the exact solution $\bar{\rho}$, although a particle at this location has already passed through the shock. The dashed dotted line in the left plot sketches what the numerical solution ρ is going to look like after some time passed. The larger $D\rho$ and $D(\rho u)$, the worse would be the effects sketched above. So we can expect, that $D\rho$ rises quickly in such a case.

However, this little example shows, that in fact all shocks, computed by the general SPH approach would pop up after some amount of time.
The question arises what we can do against this phenomenon. There are two ways.

- Introducing numerical viscosity by time integration methods. In section 2.4. we will speak about time integration methods. There we will find that we can introduce some numerical viscosity by using first order time integration schemes. This numerical viscosity is powerful enough to smooth out the instability effects sketched above. Please refer especially to section 2.4.1. to learn about the effects of numerical viscosity induced by first order time integration.

- Upwinding the numerical scheme, which is shown in chapter 6. The sketch above already suggests how to get around the popping up of shock solutions. Namely, if it would be possible to shift the function Πu to the left (in up-stream direction) by a little bit. This would directly diminish the size of the value $\frac{\partial(D\rho)}{\partial t}$ at the downstream side of the shock. If Πu is shifted far enough to the left, the mechanism of producing instabilities down-stream of the shock front is wiped out. Actually, shifting functions can be view as a method of upwinding (which means we use flow information from locations laying further upstream). The question how to upwind the general SPH scheme will be answered in chapter 6.

2.4. Time Integration of the General Smoothed Particle Scheme

We have to make some considerations about the time integration of the scheme (2.10./2.11). This scheme is obviously a system of ordinary differential equations. For time integration we can use any known ODE-solver. We shall have a distinct look at two first order methods (section 2.4.1), also at three second order methods (section 2.4.2) and a method which we hope to have advantages for solving stiff problems (i.e. when low Mach number flow is to be computed with general SPH).

2.4.1. First Order Time Integration Methods

First order methods are very simple time integration methods, because they are easy to handle. However, we shall realize that first order methods prove to provide some numerical viscosity to the smoothed particle scheme. It is necessary to analyze the numerical viscosity given by the time integration scheme in order keep the time integration stable. About the importance of the numerical viscosity, we will make a statement in subsection 2.4.1.3.

2.4.1.1. Explicit Euler Method

The most simple method to solve numerically an ODE of the form (2.18) is the explicit Euler method.

Definition 2.9: (Explicit Euler Method)
The numerical discretization of ODE (2.18) given by

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t \cdot \mathbf{G}(\ \mathbf{Y}^{(n)}\ ) \tag{2.42}$$

is called explicit Euler method (see [28]). $\mathbf{Y}^{(i)}$ is an approximation of the exact solution to the ODE (2.18) $\mathbf{Y}(t_i)$ at discrete time t_i.

This is a very fast time integration method, because it requires evaluating the function $\mathbf{G}(\ \mathbf{Y}^{(n)}\ )$ only once per time step. The numerical scheme (2.42) does not provide the exact solution to the ODE (2.18). Instead, we can establish

Lemma 2.10. The computed solution $\mathbf{Y}^{(n)}$ by the numerical discretization (2.42) is the exact solution $\mathbf{Y}(t_n)$ to some disturbed ODE of the form

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) \; - \; \left(\frac{\Delta t}{2}\right) \frac{d^2\,\mathbf{Y}}{dt^2} \; - \; \left(\frac{\Delta t^2}{6}\right) \frac{d^3\,\mathbf{Y}}{dt^3} \; + \; \mathbf{O}(\Delta t^3) \tag{2.43}$$

Proof:
The proof uses simply Taylor series expansion.

The term $-\left(\frac{\Delta t}{2}\right) \frac{d^2\,\mathbf{Y}}{dt^2}$ causes problems because it acts like a negative viscosity term. Please refer to section 2.4.1.3 where the viscous effects of this term are sketched. This negative viscous behavior of $-\left(\frac{\Delta t}{2}\right) \frac{d^2\,\mathbf{Y}}{dt^2}$ causes the computed solution to be unstable, especially when shocks occur. Hence, if using the explicit Euler method, it is absolutely necessary to introduce terms of artificial viscosity. The most easy way of doing it is to setup an explicit numerical scheme for the ODE

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) \; + \; \alpha \frac{d^2\,\mathbf{Y}}{dt^2} \tag{2.44}$$

where $\alpha \frac{d^2\,\mathbf{Y}}{dt^2}$ acts as an artificial viscosity term. The numerical scheme suggested here is given by

Definition 2.11: (Explicit Euler with Artificial Viscosity)
The numerical time integration scheme for ODE (2.18), given by

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t\, \mathbf{G}(\mathbf{Y}^{(n)}) + \alpha \;\left[\; \mathbf{G}(\mathbf{Y}^{(n)}) - \mathbf{G}(\mathbf{Y}^{(n-1)}) \;\right] \tag{2.45}$$

is the explicit Euler scheme with incorporated artificial viscosity.
Remark: The scheme given here is the direct explicit approximation of equation (2.44). Based on the scheme (2.45), we can establish

Lemma 2.12
The numerical approximation $\mathbf{Y}^{(i)}$ provided by the scheme (2.45) is the exact solution $\mathbf{Y}(t_i)$ to the perturbed ODE

$$\frac{d\,\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) + \left(\alpha - \frac{\Delta t}{2}\right) \cdot \frac{d^2\,\mathbf{Y}}{dt^2} - \left(\alpha^2 + \frac{\Delta t^2}{6}\right) \cdot \frac{d^3\,\mathbf{Y}}{dt^3} + \mathbf{O}(\Delta t^3) \tag{2.46}$$

Proof: The proof uses again Taylor series expansion.
Remark: This lemma shows that α has to be chosen such that $\left(\alpha - \frac{\Delta t}{2}\right) > 0$ in order to provide positive numerical viscosity to the particle method. We would like to remind the reader that the viscous behavior of the term $\frac{d^2\,\mathbf{Y}}{dt^2}$ is sketched in section 2.4.1.3.

2.4.1.2 Truncated Implicit Euler Method

The method which we used in many of our computations is a truncated implicit Euler method. At first let us introduce the classical implicit Euler method.

Definition 2.13. (Implicit Euler)
The numerical discretization of ODE (2.18) given by

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t \cdot \mathbf{G}(\ \mathbf{Y}^{(n+1)}\ ) \tag{2.47}$$

is called implicit Euler method. We call $\mathbf{Y}^{(n)}$ the approximate numerical solution of ODE (2.18) at the discrete time t_n, and $\Delta t = t_{n+1} - t_n$ is the time step size.

Lemma 2.14
The numerical approximation $\mathbf{Y}^{(n)}$ provided by the numerical scheme (2.48) is the exact solution $\mathbf{Y}(t_n)$ to the perturbed ODE

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) + \left(\frac{\Delta t}{2}\right) \cdot \frac{d^2\mathbf{Y}}{dt^2} - \left(\frac{\Delta t^2}{6}\right) \cdot \frac{d^3\mathbf{Y}}{dt^3} + \mathbf{O}(\Delta t^3) \tag{2.48}$$

Proof: As usual, the proof uses Taylor series expansion.

The scheme (2.47) therefore provides positive numerical viscosity (see section 2.4.1.3.) because of the term $+\left(\frac{\Delta t}{2}\right) \cdot \frac{d^2\mathbf{Y}}{dt^2}$ in (2.48). The scheme (2.47) provides another very important property, namely it is called to be A-stable (see [28], [6]). This just means that the solution $\mathbf{Y}^{(n)}$ stays bounded for any n and any Δt provided that the initial condition $\mathbf{Y}(t_0)$ is bounded. However, the implicit Euler scheme causes one difficulty. The computation of $\mathbf{G}(\ \mathbf{Y}^{(n+1)}\ )$ requires the solution at time t_{n+1} which is not known a priori. One very simple method to solve for $\mathbf{Y}^{(n+1)}$ is to iterate equation (2.47) by a fixed point method of the form

$$\mathbf{Y}_{j+1}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t \cdot \mathbf{G}(\ \mathbf{Y}_j^{(n+1)}\ ), \quad j = 0, 1, ... \tag{2.49}$$

We call $\mathbf{Y}_{j+1}^{(n+1)}$ the iterated solution after $j+1$ iterations. The closer $\mathbf{Y}_0^{(n+1)}$ to the exact solution $\mathbf{Y}^{(n+1)}$ the less iteration steps are required. Since $\mathbf{Y}^{(n+1)}$ cannot be very far away from $\mathbf{Y}^{(n)}$, we choose $\mathbf{Y}_0^{(n+1)} = \mathbf{Y}^{(n)}$.

Theorem 2.16: (Convergence of Fixed Point Iterations)
The fixed point iteration (2.49) converges in the sense

$$\lim_{j\to\infty} \|\mathbf{Y}_j^{(n+1)} - \mathbf{Y}^{(n+1)}\| = 0 \tag{2.50}$$

if there is a matrix norm $\|.\|$ such that we have

$$\Delta t \cdot \left\| \frac{\partial \mathbf{G}}{\partial \mathbf{Y}} \right\| \; < \; 1 \tag{2.51}$$

Proof: The proof of this theorem is to be found for example in [28] or [26].

The iteration scheme proposes to perform infinitely many iteration steps in order to obtain $\mathbf{Y}^{(n+1)}$, however this would not be the sense of fast computation performance. We will restrict ourself to as few as possible iteration steps. That is why we will not use the exact implicit Euler scheme but a truncated method meaning that the iteration loop is truncated after a certain number of steps.

Definition 2.17: (Truncated Implicit Euler Method)
Given the set of ODE (2.18). Given $\mathbf{Y}^{(n)}$ the computed solution at time t_n. Given $m \in \mathbf{N}$, $m > 0$. Given the truncated fixed point iteration

$$\mathbf{Y}_{j+1}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t \cdot \mathbf{G}(\; \mathbf{Y}_j^{(n+1)} \;), \quad 0 \leq j < m \tag{2.52}$$

with $\mathbf{Y}_0^{(n+1)} := \mathbf{Y}^{(n)}$. Then we define

$$\mathbf{Y}^{(n+1)} := \mathbf{Y}_m^{(n+1)} \tag{2.53}$$

For our computations we chose $m = 2$. In this case, the method reduces to a simple predictor-corrector method. Nevertheless, we obtained very good results. Choosing $m > 2$ resulted in consuming much more computation time but did not lead to considerable improvement of the solution.

As for the explicit Euler method, we are as well able to introduce an artificial viscosity term here, because the user might want to have more or even less numerical viscosity as it is invoked by the truncated implicit Euler method. Hence, we state

Definition 2.18 (Implicit Euler with artificial viscosity)
The numerical discretization of ODE (2.18) given by

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t \cdot \mathbf{G}(\mathbf{Y}^{(n+1)}) + \alpha \left(\mathbf{G}(\mathbf{Y}^{(n+1)}) - \mathbf{G}(\mathbf{Y}^{(n)}) \right) \tag{2.54}$$

is called implicit Euler method with artificial viscosity. Here, one would like to choose $\alpha < 0$ in order to reduce the numerical viscosity induced by the time integration scheme. We can also give a statement, what kind of ODE is exactly solved with this scheme.

Lemma 2.19
The numerical approximation $\mathbf{Y}^{(i)}$ provided by the scheme (2.54) is the exact solution $\mathbf{Y}(t_i)$ of the perturbed ODE

$$\frac{d\,\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) + \left(\alpha + \frac{\Delta t}{2}\right) \cdot \frac{d^2\,\mathbf{Y}}{dt^2} - \left(\alpha^2 + \alpha\,\Delta t + \frac{\Delta t^2}{6}\right) \cdot \frac{d^3\,\mathbf{Y}}{dt^3} + \mathbf{O}(\Delta t^3) \quad (2.55)$$

Remark: It is obvious to choose $\alpha > -\frac{\Delta t}{2}$ in order to keep the numerical viscosity positive.

Again, the scheme (2.54) is implicit. Therefore, it is again necessary to choose some method to solve for $\mathbf{Y}^{(n+1)}$. Similar to definition 2.17, we suggest a simple fixed point iteration.

Definition 2.20 (Truncated Implicit Euler with Artificial Viscosity)
The fixed point iteration given by

$$\mathbf{Y}_{j+1}^{(n+1)} = \mathbf{Y}^{(n)} + (\Delta t + \alpha) \cdot \mathbf{G}(\mathbf{Y}_j^{(n+1)}) - \alpha \cdot \mathbf{G}(\mathbf{Y}^{(\mathbf{n})})\ , \quad j < m \qquad (2.56)$$

with $\mathbf{Y}_0^{(n+1)} := \mathbf{Y}^{(n)}$
and $\mathbf{Y}^{(n+1)} := \mathbf{Y}_m^{(n+1)}$
for given $m \in \mathbf{N},\ m > 0$, is called a truncated scheme for implicit Euler with artificial viscosity.

Remark: The iteration stated above converges if there is a matrix norm such that $(\Delta t + \alpha)\left\|\frac{\partial \mathbf{G}}{\partial \mathbf{Y}}\right\| < 1$.

2.4.1.3. Numerical Viscosity Provided by First Order Schemes

As it was mentioned in the last two subsections, first order time integration schemes provide some numerical viscosity. We have to take care of the numerical viscosity especially when we would like to compute shocks with the general SPH scheme. If the sign of the viscous term is wrong then instabilities are likely to occur. If the sign is right, then the numerical viscosity provides smoothing. In this subsection we are going to explain how this numerical viscosity is to be understood.

As we saw, the computed solution $\mathbf{Y}^{(i)}$ is not the exact solution to the ODE

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y})$$

at time t_i, but it is rather the exact solution to some perturbed ODE of the form

$$\frac{d\mathbf{Y}}{dt} = \mathbf{G}(\mathbf{Y}) + A_{nv}\,\frac{d^2\mathbf{Y}}{dt^2} + \mathbf{O}(\Delta t^2) \tag{2.57}$$

We said previously that the term $A_{nv}\,\frac{d^2\mathbf{Y}}{dt^2}$ needs to be considered carefully.

In order to understand the term $\frac{d^2\mathbf{Y}}{dt^2}$, we first try to express it in another way, restricting ourselfs only to the one-dimensional Euler case. First, let us express the second total time derivative for the density. We have

$$\frac{d\rho}{dt} = -\rho\,\frac{\partial \Pi^n_{\omega_N} u}{\partial x}$$

with $u := v^{(1)}$, $x := x^{(1)}$. It follows that

$$\frac{d^2\rho}{dt^2} = -\frac{d\rho}{dt}\,\frac{\partial \Pi^n_{\omega_N} u}{\partial x} - \rho\,\frac{d}{dt}\left(\frac{\partial \Pi^n_{\omega_N} u}{\partial x}\right)$$

This leads to

$$\frac{d^2\rho}{dt^2} = \rho\,\left(\frac{\partial \Pi u}{\partial x}\right)^2 - \rho\left(\frac{\partial^2 \Pi u}{\partial t\,\partial x} + \Pi u\,\frac{\partial^2 \Pi u}{\partial x^2}\right)$$

which we may write down as

$$\frac{d^2\rho}{dt^2} = -\rho\,\frac{\partial^2 \Pi u}{\partial t\,\partial x} + \rho\,\left(\frac{\Pi u}{\partial x}\right)^2 - \rho \Pi u\,\frac{\partial^2 \Pi u}{\partial x^2}$$

Similar calculations lead to

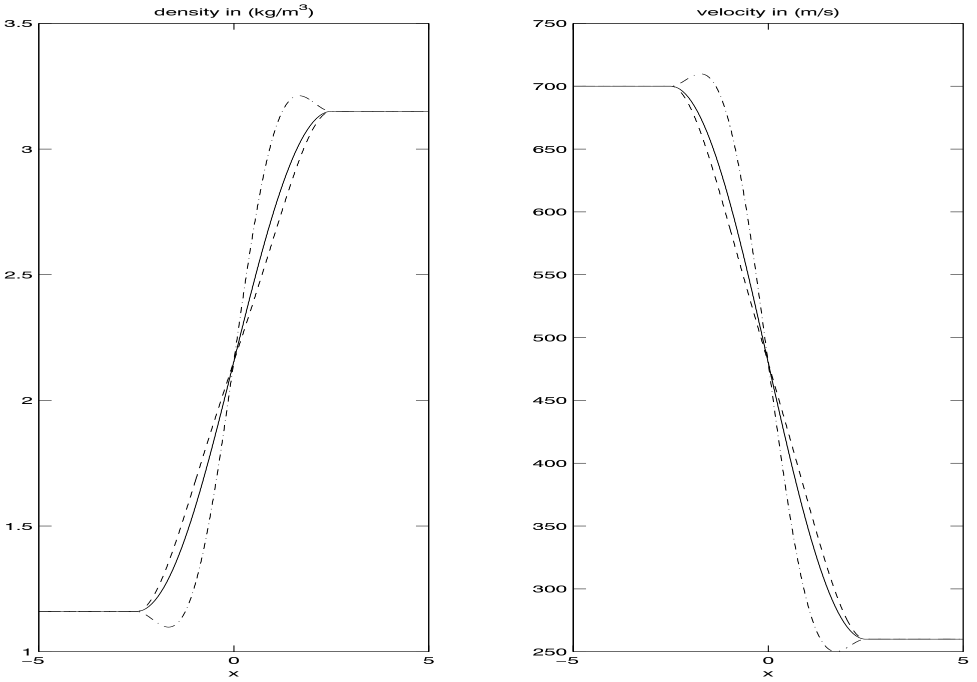
$$\begin{aligned}\frac{d^2(\rho u)}{dt^2} \;&=\; -\frac{\partial^2 \Pi p}{\partial t\,\partial x} + \frac{\partial \Pi u}{\partial x}\,\frac{\partial \Pi p}{\partial x} - \Pi u\,\frac{\partial^2 \Pi p}{\partial x^2} \;+ \\ &\quad -\,(\rho u)\,\frac{\partial^2 \Pi u}{\partial t\,\partial x} + (\rho u)\,\left(\frac{\partial \Pi u}{\partial x}\right)^2 - \,(\rho u)\,\Pi u\,\frac{\partial^2 \Pi u}{\partial x^2}\end{aligned}$$

and

$$\begin{aligned}\frac{d^2(\rho E)}{dt^2} \;&=\; -\frac{\partial^2 \Pi(pu)}{\partial t\,\partial x} + \frac{\partial \Pi u}{\partial x}\,\frac{\partial \Pi(pu)}{\partial x} - \Pi u\,\frac{\partial^2 \Pi(pu)}{\partial x^2} \;+ \\ &\quad -\,(\rho E)\,\frac{\partial^2 \Pi u}{\partial t\,\partial x} + (\rho E)\,\left(\frac{\partial \Pi u}{\partial x}\right)^2 - \,(\rho E)\,\Pi u\,\frac{\partial^2 \Pi u}{\partial x^2}\end{aligned}$$

In higher dimensions we would obtain a lot more terms similar to the ones above. The terms we obtained are hard to be interpreted as Navier-Stokes like viscosity terms, some of them do remember to Navier-Stokes, some others do not. Unfortunately, the equations above do not help us very much to understand the role of the term $A_{nv}\,\frac{d^2\mathbf{Y}}{dt^2}$.

However, we succeed to give a graphical interpretation on how the term $A\,\frac{d^2\mathbf{Y}}{dt^2}$ is acting. Let us illustrate the importance of numerical viscosity by having a look at the following stationary shocks in 1D computed with the general smoothed particle scheme (solid lines).



The particles in this example run from left to right. A particle with index i, entering the shock (for $x < 0$), would experience

$$\frac{d^2\rho_i}{dt^2} > 0$$
$$\frac{d^2v_i^{(1)}}{dt^2} < 0$$

The same particle, when leaving the shock (for $x > 0$) would experience

$$\frac{d^2\rho_i}{dt^2} < 0$$
$$\frac{d^2v_i^{(1)}}{dt^2} > 0$$

Let us first suppose $A_{nv} < 0$. Then the term $A_{nv}\,\frac{d^2\rho}{dt^2}$ is a negative contribution to $\frac{d\rho_i}{dt}$

of a particle entering the shock (meaning that $\frac{d\rho_i}{dt}$ is smaller and so, ρ_i rises not as quickly). $A_{nv}\,\frac{d^2\rho}{dt^2}$ is a positive contribution to $\frac{d\rho_i}{dt}$ of the same particle when it leaves the shock (hence ρ_i rises faster). By this mechanism, the shock region is becoming more narrow which, by itself, amplifies this mechanism. So, instabilities are likely to occur. An illustration of this effect is seen in the plots above (dashed-dotted lines).

Now suppose $A_{nv} > 0$. The the effects turn their sign around, the evolution of the density (velocity) is accelerated for those particles entering the shock and decelerated for those ones leaving the shock. Hence, the shock is about to smooth out. (See dashed line in the plots above).

We have just showed to have necessarily $A_{nv} \geq 0$ when shocks occur. By experiment we found out that positive numerical viscosity is also capable to smooth out instabilities that may occur due to effects that are described in section 2.3.3. We did not show how exactly the value for A_{nv} has to be chosen. By analysis, we did not find out how to set A_{nv}. It remained a trial-and-error procedure to hit the right numerical viscosity.

If one does not use the upwinding technique shown in chapter 6, then by trial-and-error, we found out that the method produced fine results for shocks when choosing

$$A_{nv} = (0.1...0.2) \cdot \frac{h}{c} \tag{2.58}$$

Here, h is the smoothing length and c is the speed of sound. Please note that $\frac{h}{c}$ is the amount of time the sound would need to travel the distance of h.

If the upwinding technique is chosen, then one clearly can use $A_{nv} = 0$ because then the viscosity effects are produced by the upwind scheme.

Conclusion: When choosing a first order time integration method, we have to make sure that the numerical viscosity remains positive (possibly by adding artificial viscosity). This also means that no time integration method of higher order seems to be usable for the general smoothed particle method. That is because a time integration method of higher order would provide zero numerical viscosity. In order to indeed employ higher order methods, it is necessary to employ the technique of upwinding the general smoothed particle scheme, thus involving viscosity not by time integration, but by the particle scheme itself.

In the next sections, we are going to exhibit higher order time integration methods, which may be employed provided that the particle scheme is an upwind scheme.

2.4.2. Second Order Methods

A second order method is required when we do not wish to have numerical viscosity involved through time integration. This might, for example, be necessary when we would like to compute Navier-Stokes flow with a small fluid-mechanical viscosity (i.e. if the numerical viscosity would wipe out all effects of physical viscosity). Especially upon the occurance of shocks, second order methods require that the general smoothed particle scheme be subject to some upwind procedure. Please refer to chapter 6 to learn about how to set up an upwind method for general SPH.

In order to keep this section short, we restrict ourselves to exhibiting three second order methods.

2.4.2.1. Second Order Explicit Runge-Kutta Method

A method very simple to set up into practice to find solutions to ODE (2.18) is a second order explicit Runge-Kutta method given by

$$\mathbf{Y}^{(n+\frac{1}{2})} = \mathbf{Y}^{(n)} + \frac{\Delta t}{2}\,\mathbf{G}(\mathbf{Y}^{(n)}) \tag{2.59}$$

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \Delta t\,\mathbf{G}(\mathbf{Y}^{(n+\frac{1}{2})}) \tag{2.60}$$

A lot of the computation examples in the appendix were run using this method.

2.4.2.2. Second Order Implicit Method with Truncation

Another second order method is the implicit scheme

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \frac{\Delta t}{2}\left(\mathbf{G}(\mathbf{Y}^{(n+1)}) + \mathbf{G}(\mathbf{Y}^{(n)})\right) \tag{2.61}$$

This scheme has the advantage that it is A-stable. It has the disadvantage that it is implicit. Again, as for the implicit Euler method (refer to section 2.4.1.2) we use a fixed point iteration of the form

$$\mathbf{Y}_{i+1}^{(n+1)} = \mathbf{Y}^{(n)} + \frac{\Delta t}{2}\left(\mathbf{G}(\mathbf{Y}_i^{(n+1)}) + \mathbf{G}(\mathbf{Y}^{(n)})\right) \quad 0 \le i < m \tag{2.62}$$

where we set

$$\mathbf{Y}_0^{(n+1)} := \mathbf{Y}^{(n)}$$

and where we define

$$\mathbf{Y}^{(n+1)} := \mathbf{Y}_m^{(n+1)} \tag{2.63}$$

In practice we used $m = 2$.

2.4.2.3. First Order Method with Zero Numerical Viscosity

A second order time integration method is also obtained, when we choose the explicit Euler scheme with artificial viscosity (refer to equation (2.45)) and set the free factor $\alpha = \frac{\Delta t}{2}$. In this case, the numerical scheme is

$$\mathbf{Y}^{(n+1)} = \mathbf{Y}^{(n)} + \frac{3}{2}\Delta t\, \mathbf{G}(\mathbf{Y}^{(n)}) - \frac{1}{2}\Delta t \mathbf{G}(\mathbf{Y}^{(n-1)})$$

It redevelops to a multi step method. This, as equation (2.46) shows, gives also a time integration scheme of second order, since $A_{nv} = \alpha - \frac{\Delta t}{2} = 0$. This scheme has one advantage in comparison to the other two exhibited time integration methods, namely it requires only **one** evaluation of the function $\mathbf{G}(\mathbf{Y})$ for one time step. The others, as we quickly see, require at least **two** function evaluations. However, this method has one disadvantage, namely it requires the value $\mathbf{G}\left(\mathbf{Y}^{(n-1)}\right)$. The question arises what to do with those particles that come into play at time t_n, for example particles introduced at inflow boundaries. For them, no value for $\mathbf{G}\left(\mathbf{Y}^{(n-1)}\right)$ is known. The treatment of newly added particles is therefore an important question.

There was no research done on this question yet, but we consider it as to be worth while since this could be a very fast time integration method of second order, compared to the first ones given here.

2.4.3. A Time Integration Method for Low Mach Number Flow

For low Mach number flow the ODE (2.18) becomes very stiff, because the difference between the sound speed and the velocity of the fluid is considerably large. For the first order and second order methods, we usually have restrictions to the time step which are

$$\Delta t < k \cdot \frac{h}{c} \tag{2.64}$$

where $k \in \mathbf{R},\ k < 1$ is a constant, h is the smoothing length and c is the speed of sound. The proof of equation (2.64) is omitted here. It is very straight forward, but also very long and tedious.

Now we would like to exhibit a method, which does not set restrictions to the time step size. Let $\mathbf{Y}^{(n)}$ be the computed solution at time t_n. We linearize the given ODE (2.18) by

$$\frac{d\mathbf{Y}}{dt} \;=\; \mathbf{G}(\mathbf{Y}^{(n)}) \;+\; \frac{\partial \mathbf{G}}{\partial \mathbf{Y}}(\mathbf{Y}^{(n)}) \cdot \left(\, \mathbf{Y} - \mathbf{Y}^{(n)} \,\right) \tag{2.65}$$

For simplicity we write equation (2.65) as

$$\frac{d\mathbf{Y}}{dt} \;=\; \mathbf{A} \;+\; \mathbf{B} \cdot \mathbf{Y} \tag{2.66}$$

where
$\mathbf{A} := \mathbf{G}(\mathbf{Y}^{(n)}) - \mathbf{B} \cdot \mathbf{Y}^{(n)}$ and
$\mathbf{B} := \frac{\partial \mathbf{G}}{\partial \mathbf{Y}}$

It is easy to verify, that the exact solution to the linearized ODE (2.66) is given by

$$\begin{aligned} \mathbf{Y}_{\mathrm{lin}}^{(n)}(t) \quad &= \quad \left(\mathbf{I} \;+\; \frac{1}{2!}\mathbf{B}\Delta t \;+\; \frac{1}{3!}(\mathbf{B}\Delta t)^2 \;+\; \frac{1}{4!}(\mathbf{B}\Delta t)^3 \;+ \ldots \right) \cdot \mathbf{A}\,\Delta t \,+ \\ &\quad\;\; \left(\mathbf{I} \;+\; \frac{1}{1!}\mathbf{B}\Delta t \;+\; \frac{1}{2!}(\mathbf{B}\Delta t)^2 \;+\; \frac{1}{3!}(\mathbf{B}\Delta t)^3 \;+\; \ldots \right) \cdot \mathbf{Y}^{(n)} \end{aligned} \tag{2.67}$$

where
$\Delta t := t - t_n$.
We are now in the situation to compute the exact solution $\mathbf{Y}_{\mathrm{lin}}^{(n)}(t)$ to the linearized ODE (2.65) in the time interval $[t, t_{n+1}]$. If we fix a certain time step $\Delta t := t_{n+1} - t_n$, then we define the time integration through linearization by

$$\mathbf{Y}^{(n+1)} := \mathbf{Y}_{\mathrm{lin}}^{(n)}(t_{n+1}) \tag{2.68}$$

$\mathbf{Y}^{(n+1)}$ is easily computed if it is possible to compute $\mathbf{B}$ accurately enough and if it is possible to store it efficiently in computer memory. In addition, we require a criterion to truncate the summation series.

Big advantages:
1.) There is no restriction to the time step size $\Delta t = t_{n+1} - t_n$ because we compute the exact solution to the given ODE (2.65). Since the ODE (2.65) itself is linear, the solution $\mathbf{Y}_{\mathrm{lin}}^{(n)}(t)$ stays bounded for all $t < \infty$ if $\mathbf{A}$ and $\mathbf{B}$ are bounded.

2.) A multiplication of $\mathbf{B}$ with a vector takes much less computation time than the evaluation of the function $\mathbf{G}(\mathbf{Y})$. So, the hope is that we save a lot of computation time by employing the method as described above, because we can choose the time step much larger than for the methods described in the previous sections.

Remark: No research is done on this time integration method for low Mach number flow yet. However, we consider it as to be one very important point in future research for general smoothed particle methods. In fact, we tested this method for 1D cases and it worked fine. This gives us hope that it could also work fine for 2D/3D. Up to now, we did not set it into practice.

Chapter 3

3. Exhibition of a Class of Smoothed Particle Operators

In chapter 2 we introduced a new numerical scheme for a general smoothed particle operator Π^n. As we discussed there, it was absolutely necessary to come up with such a new scheme in order to use arbitrary smoothed particle operators. Especially we would like to use those operators, which provide convenient approximation properties at the boundaries. In chapter 2, we did not mention yet, how to establish them. However, in the present chapter we will turn to establishing Π^n-operators in detail. As we shall see, we are still going to use a sort of smoothing kernel, distributing weights on the particles surrounding any arbitrary viewpoint. Thus, similarly to classical SPH, we are going to work with a smoothing length h representing the interaction radius between particles.

3.1. Construction of Smoothed Particle Operators of Different Orders

In this section we will exhibit one very practicable method of setting up a Π^n-operator. This does not mean that there are no other ways to do it, but the method introduced below is successful since it worked very stable for the new particle scheme and it demonstrates excellent approximation properties at the boundaries. The method presented is based on a local least squares approximation of a given function defined on a set of particles. Least squares approximations were already used in SPH models, for example by G. Dilts (see [7]) who used least squares techniques for an SPH approach for impact mechanics. The employment of least squares techniques in SPH is commonly referred to as 'moving least squares'. So, what is presented here is nothing new, but it is the attempt to employ the least squares idea for the general SPH model given in chapter 2.

We start with defining the set of all polynomials of degree d in ν dimensions by

$$\mathcal{P}^d := \left\{ p : \mathbf{R}^\nu \to \mathbf{R} \mid p(\mathbf{x}) := \sum_{|\alpha| \le d} C_\alpha \, \mathbf{x}^\alpha \; ; \quad C_\alpha \in \mathbf{R}; \quad \mathbf{x} \in \mathbf{R}^\nu \right\} \tag{3.1}$$

where
$\alpha := (\alpha^{(1)}, ..., \alpha^{(\nu)})$ where $\alpha^{(i)} \in \mathbf{N}_0$,
$\mathbf{x}^\alpha := \left(x^{(1)}\right)^{\alpha^{(1)}} \cdot ... \cdot \left(x^{(\nu)}\right)^{\alpha^\nu}$, and
$|\alpha| := \alpha^{(1)} + ... + \alpha^{(\nu)}$.

We define the series of α by
$\alpha_0 = (0, ..., 0)$, $\alpha_1 = \mathbf{e}_1 = (1, 0, ..., 0)$, $\alpha_2 = \mathbf{e}_2 = (0, 1, ..., 0)$, etc.

We will now define a local least squares approximation of some arbitrary function f. The idea is to find local polynomials of degree d which fit f best in a least squares sense.

Definition 3.1. (Local Least Squares Approximation of Different Orders)

Let $\Omega(t) \subset \mathbf{R}^\nu$ be the domain of fluid flow and $\omega_N(t)$ be a particle system being established in $\Omega(t)$. Let $\mathbf{f} : T \to \mathbf{R}^N$ be an arbitrary vector of discrete function values with $\mathbf{f}(t) := (\ f_1(t), ..., f_N(t)\ )$.
Let $W^n : \mathbf{R}^+ \to \mathbf{R}^+$ be a weighting function with the properties
1.) $0 \le W^n(r) \le 1 \ \forall r \in \mathbf{R}^+$
2.) $W^n(r) = 0$ for $r > 1$
3.) $W^n \in C^n(\mathbf{R}^+)$
4.) $\frac{dW^n}{dr} = 0$ if $r = 0$

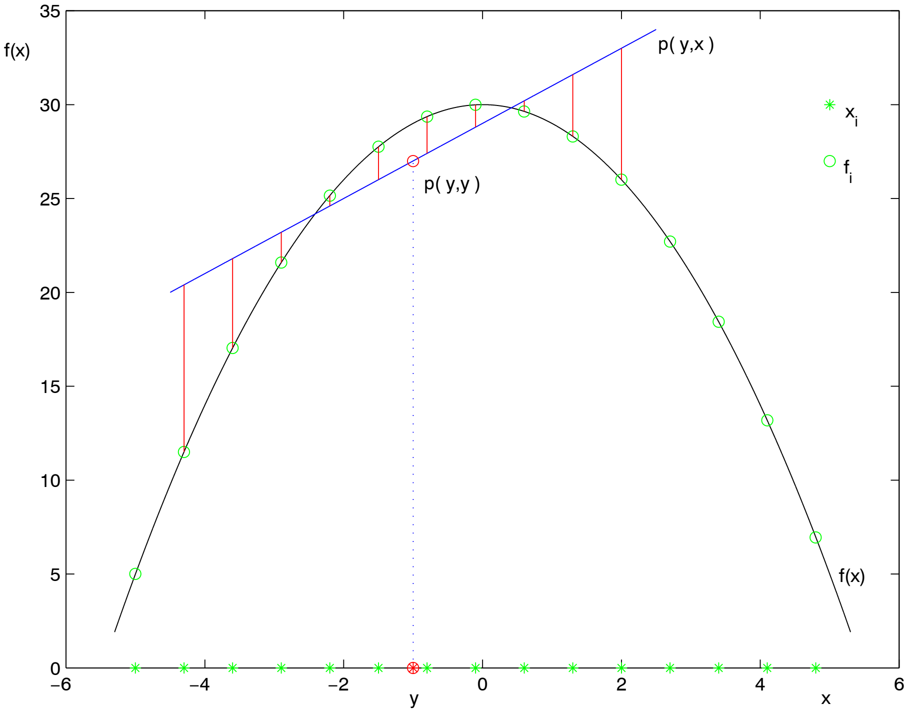
Let $h : \bar{\Omega} \to \mathbf{R}^+$ be the local smoothing length (also referred to as local interaction radius between two particles).
Let $p_\mathbf{y} \in \mathcal{P}^d$.
At some fixed time $\bar{t}$, we define $r(\mathbf{y}, \mathbf{x}) := \left(\frac{\|\mathbf{y}-\mathbf{x}\|_2}{h(\bar{t}, \mathbf{x})} \right)^2$, which is the normalized distance between the points $\mathbf{y}, \mathbf{x} \in \Omega(\bar{t})$.
We call $p_\mathbf{y}$ a least squares approximation of order d of the function values $\mathbf{F}$ in the neighborhood of $\mathbf{y} \in \Omega(\bar{t})$ if it minimizes the functional

$$D = \sum_{i=1}^{N} |\ W^n(\, r(\mathbf{y}, \mathbf{x}_i)\,)\ m_i\ (\, f_i(\bar{t}) - p_\mathbf{y}(\mathbf{x}_i)\,)|^2 \tag{3.2}$$

For the purpose of further reference we introduce another form of writing down the least squares approximation polynomial

$$p(\mathbf{y},\mathbf{x}) := p_{\mathbf{y}}(\mathbf{x}) \tag{3.3}$$

Remark 1: Definition 3.1 states that we are simply looking for a polynomial of degree d approximating the function values $\mathbf{f}$ locally at the point $\mathbf{y}$ by making a possibly small error. The distances between $p_{\mathbf{y}}(\mathbf{x}_i)$ and $f_i(\bar{t})$ are weighted by some weighting function W^n such that errors made far away from $\mathbf{y}$ have less importance than errors made close by. The following sketch is a simple visualization of definition 3.1 for a one-dimensional case.



The least squares task is to place the polynomial such that the sum of the (weighted) squared lengths between the discrete function values $f_i(\bar{t})$ and the values of the polynomial $p(\mathbf{y},\mathbf{x}_i)$ (red marked lines in the plot above) is minimized.

Remark 2: We do not require the smoothing length to be constant throughout $\Omega(t)$, hence all further considerations are done assuming, that the smoothing length can vary locally.

Remark 3: The weights m_i have, unlike in classical SPH, not the meaning of mass. Here, the m_i are 'physical' weights, telling how much important a particle is for the establishment of the least squares approximation. Please refer to section 5.1, where the usage of the weights m_i is lined out in detail.

The practical computation of $p(\mathbf{y}, \mathbf{x})$ is very simple. It consists in finding the coefficients $C_\alpha(\mathbf{y})$ which are indeed dependent on the position $\mathbf{y}$ where the polynomial has to be established. In order to compute $C_\alpha(\mathbf{y})$, we rewrite (3.2) in matrix vector formulation

$$D = \|\mathbf{A} \cdot \mathbf{C} - \mathbf{F}\|_2$$

with

$$\mathbf{A}^{(i,j)} \; = \; W_i \cdot (\mathbf{y} - \mathbf{x}_i)^{\alpha_j}$$

$$\mathbf{C} \; = \; [C_{(000)}, C_{(100)}, C_{(010)}, ...]^T \; = \; [C_{\alpha_0}, C_{\alpha_1}, C_{\alpha_2}, ...]^T$$

$$\mathbf{F}^{(i)} \; = \; W_i \cdot f_i$$

where
$W_i := W^n(r(\mathbf{y}, \mathbf{x}_i))\, m_i$

The minimizer of D is given by

$$\mathbf{C} = (\mathbf{A}^T \mathbf{A})^{-1} \cdot (\mathbf{A}^T \mathbf{F})$$

(refer to [26]).

Convention: We say that there exists a unique least squares approximation $p(\mathbf{y}, \mathbf{x})$ at point $\mathbf{y}$ if the inverse $(\mathbf{A}^T \mathbf{A})^{-1}$ exists.

We have to ask, for what cases $(\mathbf{A}^T \mathbf{A})$ is invertible. The invertibility depends certainly on the degree of the polynomials d and of the positions $\mathbf{x}_i$ of the particles surrounding $\mathbf{y}$. For $d = 0$ we can say that $(\mathbf{A}^T \mathbf{A})$ is invertible if there is at least one particle in the neighborhood of $\mathbf{y}$. For $\nu = 3$ and $d = 1$ we can say, that $(\mathbf{A}^T \mathbf{A})$ is invertible if there are at least 4 particles in the neighborhood of $\mathbf{y}$ which are not all placed in a plane. For $d = 2$ we need at least 10 surrounding particles which should not all be placed in a plane or a sphere.

Remark: As a matter of fact, we have no influence on the particle positions since the particles are moved due to the fluid velocity. For the remaining part of this chapter we assume that at any time t the particle ensemble $\omega_N(t)$

is always established such that for all $\mathbf{y} \in \Omega(t)$ there exists a local least squares approximation. Hence, for this part, we do not have to worry about the question if there are enough particles to compute the least squares approximation. In chapter 5 we shall answer the question what to do when the condition number of the matrix $(\mathbf{A}^T\mathbf{A})$ becomes bad.

Now, we would like to go on in our context. With definition 3.1 we are in the situation to give a definition for a smoothed particle operator on a function f. For this purpose, we are going to employ the local least squares approximation.

Definition 3.2 (General Smoothed Particle Operator Π^n)
At some fixed time $\bar{t}$, let $\Omega(\bar{t}) \subset \mathbf{R}^\nu$ be the domain of fluid flow and $\omega_N(\bar{t})$ be a particle system being established in $\Omega(\bar{t})$. Let $\mathbf{f} : T \to \mathbf{R}^N$ be an arbitrary vector containing discrete function values and let $p(\mathbf{y}, \mathbf{x})$ be the least squares approximation polynomial in the sense of definition 3.1 at an arbitrary point $\mathbf{y} \in \Omega(\bar{t})$ where we would like to compute the smoothed particle operator. We define

$$\Pi^n_{\omega_N} f(\bar{t}, \mathbf{y}) := p(\mathbf{y}, \mathbf{y}) \tag{3.4}$$

Convention: In this chapter, if a fixed time $\bar{t}$ is considered, we abbreviate $\Pi^n_{\omega_N} f(\bar{t}, \mathbf{y}) := \Pi^n_{\omega_N} f(\mathbf{y})$.

Remark: The definition 3.2 means nothing else than establishing the least squares polynomial in the point $\mathbf{y}$ and then assigning the value of the approximating polynomial at the position $\mathbf{y}$ to $\Pi^n_{\omega_N} f$.

Since we deal with a least squares approach, we can imagine that the approximation properties of $\Pi^n_{\omega_N} f$ are very good, even at $\partial\Omega$. For our computations, we chose $d \in \{0, 1, 2\}$.

The smoothed particle operator established in this way, does clearly belong to the set of all smoothed particle operators as defined in definition 2.1. In definition 2.1, certain properties of Π^n were required. Concerning those, it is trivial to show that

- $\Pi^n_{\omega_N}(c \cdot f) = c \cdot \Pi^n_{\omega_N} f \ , \quad c \in \mathbf{R}$
- $\Pi^n_{\omega_N}(f + g) = \Pi^n_{\omega_N} f + \Pi^n_{\omega_N} g$

and it is also clear that by the use of $W^n(r(\mathbf{y}, \mathbf{x}_i))$ we have

- $\frac{\partial}{\partial \mathbf{x}_i} \Pi^n_{\omega_N} f(t, \mathbf{y})$ is continuous and
  $\frac{\partial}{\partial \mathbf{x}_i} \Pi^n_{\omega_N} f(t, \mathbf{y}) = 0$ if $\|\mathbf{y} - \mathbf{x}_i\|_2 > h(\mathbf{x}_i)$

- $\frac{\partial}{\partial f_i}\Pi^n_{\omega_N} f(t,\mathbf{y})$ is continuous and
  $\frac{\partial}{\partial f_i}\Pi^n_{\omega_N} f(t,\mathbf{y}) = 0$ if $\|\mathbf{y}-\mathbf{x}_i\|_2 > h(\mathbf{x}_i)$

It is left to show that $\Pi^n_{\omega_N} f$ is indeed $\in C^n(\Omega(\bar{t}))$. This very important property will be shown in theorem 3.4.
Another even more important result is theorem 3.3 where we reveal some approximation properties of the smoothed particle approximation introduced in this chapter.

Theorem 3.3 (Approximation Properties of $\Pi^n_{\omega_N} f$)
Let $\Omega(t)$ be, as usual, the domain of fluid flow and let $\mathbf{y} \in \Omega(t)$. At some fixed time $\bar{t}$, let f be an analytic function in $\Omega(\bar{t})$, i.e. in the neighborhood of $\mathbf{y}$ it has a Taylor expansion of the form

$$f(\mathbf{x}) = \sum_{\alpha} B_\alpha \, (\mathbf{y}-\mathbf{x})^\alpha \quad , \quad f \in C^{d+1}(\Omega(t))$$

Let $h : \bar{\Omega} \to \mathbf{R}^+$ be the smoothing length, and let d denote the degree of the polynomial which is employed to compute $\Pi^n_{\omega_N} f$. Let $\omega_N(\bar{t})$ be a particle ensemble in $\Omega(\bar{t})$ such that $\forall \mathbf{y} \in \Omega(\bar{t})$ there exists a unique least squares approximation $p(\mathbf{y},\mathbf{x})$ (which is equivalent to supposing that, locally, there are always enough particles distributed around $\mathbf{y}$ to compute the local least squares approximation). Then we have

$$\left| \, \Pi^n_{\omega_N} f(\mathbf{y}) - f(\mathbf{y}) \, \right| \; = \; \mathbf{O}(h^{d+1}) \tag{3.5}$$

where $h := \max\limits_i \left\{ \, h(\bar{t},\mathbf{x}_i) \mid \frac{\|\mathbf{y}-\mathbf{x}_i\|_2}{h(\bar{t},\mathbf{x}_i)} < 1 \, \right\}$ is a local representative smoothing length.

Proof:
The polynomial $p(\mathbf{y},\mathbf{x})$ which is the least squares approximation of $f(\mathbf{x})$ in $\mathbf{y}$ has the expansion:

$$p(\mathbf{y},\mathbf{x}) = \sum_{|\alpha|\le d} C_\alpha \, (\mathbf{y}-\mathbf{x})^\alpha \tag{3.6}$$

Establishing $p(\mathbf{y},\mathbf{x})$ now consists in finding factors C_α such that the functional

$$D = \sum_{i=1}^{N} \left| \, W(\, r(\mathbf{y},\mathbf{x}_i)\,)\; m_i \; \left(\sum_{|\alpha|\le d} C_\alpha \, (\mathbf{y}-\mathbf{x}_i)^\alpha \; - \; \sum_{\alpha} B_\alpha \, (\mathbf{y}-\mathbf{x}_i)^\alpha \right) \right|^2$$

is minimized. This is equivalent to

$$D = \sum_{i=1}^{N} \left| \, W(\, r(\mathbf{y},\mathbf{x}_i)\,)\; m_i \; \left(\sum_{|\alpha|\le d} K_\alpha \, (\mathbf{y}-\mathbf{x}_i)^\alpha \; - \; \sum_{|\alpha|> d} B_\alpha \, (\mathbf{y}-\mathbf{x}_i)^\alpha \right) \right|^2$$

where we have to find the factors $K_\alpha := C_\alpha - B_\alpha$ such that D is minimized. The minimizing problem might also be written in matrix vector form

$$\|\mathbf{W} \cdot \mathbf{M} \cdot \mathbf{K} - \mathbf{W} \cdot \mathbf{F}\|_2 \overset{!}{=} min \tag{3.7}$$

where

$$\mathbf{W} = \begin{bmatrix} W_1 & 0 & 0 & \cdots & 0 \\ 0 & W_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & W_N \end{bmatrix} , \quad W_i = W^n(\, r(\mathbf{y}, \mathbf{x}_i)\,)\, m_i$$

$$\mathbf{M}^{(i,j)} = (\mathbf{y} - \mathbf{x}_i)^{\alpha_j}$$

$$\begin{aligned} \mathbf{K} &= \left[K_{(000)}, K_{(100)}, K_{(010)}, ...\right]^T \\ &= \left[K_{\alpha_0}, K_{\alpha_1}, K_{\alpha_2}, ...\right]^T \end{aligned}$$

$$\mathbf{F}^{(i)} = \sum_{|\alpha|>d} B_\alpha \, (\mathbf{y} - \mathbf{x}_i)^\alpha$$

The solution **K** to equation (3.7) is simply given by

$$\mathbf{K} = \left(\mathbf{M}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{M}\right)^{-1} \cdot \left(\mathbf{M}^T \cdot \mathbf{W}^T\right) \cdot (\mathbf{W} \cdot \mathbf{F}) \tag{3.8}$$

(see [26]). The occurring matrix is invertible because we supposed unique existence of the least squares approximation $p(\mathbf{y}, \mathbf{x})$. We remember that we would like to give an estimation of $|\Pi^n_{\omega_N} f(\mathbf{y}) - f(\mathbf{y})|$. However, in our context we simply have

$$\Pi^n_{\omega_N} f(\mathbf{y}) - f(\mathbf{y}) = K_{(000)} = K_{\alpha_0} \tag{3.9}$$

Hence, in order to give an estimation of $|\Pi^n_{\omega_N} f(\mathbf{y}) - f(\mathbf{y})|$, we have to give an estimation of $|K_{\alpha_0}|$. This is easily done by

$$|K_{\alpha_0}| \leq \mathcal{S} \cdot \|\mathbf{W} \cdot \mathbf{F}\|_\infty \tag{3.10}$$

where

$$\mathcal{S} := \left\| \left[\, \left(\mathbf{M}^T \cdot \mathbf{W}^T \cdot \mathbf{W} \cdot \mathbf{M}\right)^{-1} \cdot \left(\mathbf{M}^T \cdot \mathbf{W}^T\right) \,\right]^{(1,\cdot)} \right\|_1$$

is nothing else than the norm of the first row vector of the appearing matrix. It is clear, that $\mathcal{S}$ depends only on the particle positions and of the choice of W^n. For $\|\mathbf{W}\cdot\mathbf{F}\|_\infty$ we are able to derive the following estimation

$$\begin{aligned}\|\mathbf{W}\cdot\mathbf{F}\|_\infty &= \max_i\left(\left|\sum_{|\alpha|>d} W_i\,B_\alpha\,(\mathbf{y}-\mathbf{x}_i)^\alpha\right|\right) = \\ &\max_i\left(\left|\sum_{|\alpha|=d+1} W_i\,B_\alpha\,(\mathbf{y}-\mathbf{x}_i)^\alpha + \sum_{|\alpha|>d+1} W_i\,B_\alpha\,(\mathbf{y}-\mathbf{x}_i)^\alpha\right|\right)\end{aligned}$$

Since $|W_i|\leq 1$, we have immediately

$$|\,W_i\,B_\alpha\,(\mathbf{y}-\mathbf{x}_i)^\alpha\,| \;\leq\; |B_\alpha|\cdot h_i^{|\alpha|} \quad,\quad h_i = h(\mathbf{x}_i)$$

and therefore we obtain

$$\begin{aligned}\|\mathbf{F}\|_\infty &\leq \left(\sum_{|\alpha|=d+1}|B_\alpha|\,h^{d+1} + \sum_{|\alpha|>d+1}|B_\alpha|\,h^{d+2}\right) \Rightarrow \\ \|\mathbf{F}\|_\infty &= \mathbf{O}\left(\left[\sum_{|\alpha|=d+1}|B_\alpha|\right]h^{d+1}\right) \Rightarrow\end{aligned}$$

where $h = \max_i\{h_i \mid \frac{\|\mathbf{y}-\mathbf{x}_i\|_2}{h_i} < 1\}$.

With this and with (3.9) and (3.10), we indeed obtain

$$\begin{aligned}|\Pi^n_{\omega_N}f(\mathbf{y}) - f(\mathbf{y})| &= |K_{\alpha_0}| = \mathbf{O}\left(\mathcal{S}\cdot\left[\sum_{|\alpha|=d+1}|B_\alpha|\right]h^{d+1}\right) \Rightarrow \\ |\Pi^n_{\omega_N}f(\mathbf{y}) - f(\mathbf{y})| &= \mathbf{O}(h^{d+1})\end{aligned}$$

and thus, the proof is complete.

Remark: We did not say anything about the order of magnitude of $\mathcal{S}$. If particles are placed in a regular grid, then simple, but tedious computations show that $\mathcal{S} = \mathbf{O}(1)$. However, $\mathcal{S}$ is very sensible to the particle distribution around the point $\mathbf{y}$, so it can become very large. $\mathcal{S}$ will rise to infinity, as the matrix $\mathbf{A}^T\mathbf{A}$ becomes singular, i.e. if no local least squares approximation exists. It remains to hope, that the particle distribution is alway such that $\mathcal{S}$ is possibly small.

So far, we have shown local approximation properties of $\Pi^n_{\omega_N}f$. Of quite large importance is that $\Pi^n_{\omega_N} f$ be smooth. If it were not smooth, we could not compute necessary derivatives like $\nabla \Pi^n_{\omega_N} f$ or $\nabla^2 \Pi^n_{\omega_N} f$. Moreover, our considerations about conservation properties in chapter 2 would not hold. For this reason, in the following theorem we show properties of smoothness of $\Pi^n_{\omega_N} f$.

Theorem 3.4. (Smoothness of $\Pi^n_{\omega_N} f$)
At some fixed time $\bar{t}$, let $\omega_N(\bar{t})$ be a particle system such that there exists a unique least squares approximation $p(\mathbf{y}, \mathbf{x})$ $\forall \mathbf{y} \in \Omega(t)$. Let $f : \Omega(\bar{t}) \to \mathbf{R}$ be an arbitrary function. Let W^n be the weighting function used to establish $\Pi^n_{\omega_N} f$. Then

$$\Pi^n_{\omega_N} f \in C^n(\Omega(t))$$

Proof:
We consider the point $\mathbf{y} \in \Omega(\bar{t})$. In order to establish the least squares approximation of f in $\mathbf{y}$ we have to minimize the functional

$$D = \sum_{i=1}^{N} \left| W_i \left(\sum_{|\alpha| \le d} C_\alpha \, (\mathbf{y} - \mathbf{x}_i)^\alpha \; - \; f(\mathbf{x}_i) \right) \right|^2$$

where $W_i := W^n(\; r(\mathbf{y}, \mathbf{x}_i) \;) \cdot m_i$
As already shown, written in matrix vector form, the functional appears as

$$D = (\; \| \, \mathbf{A} \cdot \mathbf{C} - \mathbf{F} \, \|_2 \;)^2$$

where we have to find the vector $\mathbf{C}$ such that D is minimized. Similar to the proof of theorem 3.3, $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{F}$ look like

$$\mathbf{A}^{(i,j)} \; = \; W_i \cdot (\mathbf{y} - \mathbf{x}_i)^{\alpha_j}$$

$$\begin{aligned} \mathbf{C} \;&=\; \left[C_{(000)}, C_{(100)}, C_{(010)}, ... \right]^T \\ &=\; \left[C_{\alpha_0}, C_{\alpha_1}, C_{\alpha_2}, ... \right]^T \end{aligned}$$

$$\mathbf{F}^{(i)} \; = \; W_i \cdot f(\mathbf{x}_i)$$

The solution for $\mathbf{C}$ is given by the linear system

$$\mathbf{A}^T \mathbf{A} \cdot \mathbf{C} = \mathbf{A}^T \mathbf{F} \tag{3.11}$$

(see [26]). We immediately realize, that

$$\Pi^n_{\omega_N} f(\mathbf{y}) = C_{\alpha_0}(\mathbf{y}) \tag{3.12}$$

In order to show smoothness of $\Pi^n_{\omega_N} f$, it suffices to show smoothness of $\mathbf{C}$. Thus, we shall show by induction, that $\mathbf{C}$ is n times continuously differentiable.

1st step: We show that $\mathbf{C}$ is continuous:
Since we assumed that a unique least squares approximation $p(\mathbf{y}, \mathbf{x})$ exist for all $\mathbf{y} \in \Omega(t)$, the matrix $\mathbf{A}^T\mathbf{A}$ must be invertible $\forall \mathbf{y} \in \Omega(t)$. Moreover $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{F}$ are continuous, because the smoothing kernel W^n is continuous. But then $\mathbf{C}$ must also be continuous $\forall \mathbf{y} \in \Omega(t)$.

2nd step: We show that $\frac{\partial^k \mathbf{C}}{\partial y^k_{(j)}}$ is continuous, if $\frac{\partial^0 \mathbf{C}}{\partial y^0_{(j)}}$ ... $\frac{\partial^{k-1} \mathbf{C}}{\partial y^{k-1}_{(j)}}$ are also continuous and $k \leq n$:
The k-th derivative of equation (3.12) is

$$\sum_{i=0}^{k-1} \left(\begin{array}{c} i \\ k \end{array} \right) \frac{\partial^{k-i}(\mathbf{A}^T\mathbf{A})}{\partial y^{k-i}_{(j)}} \cdot \frac{\partial^i \mathbf{C}}{\partial y^i_{(j)}} + (\mathbf{A}^T\mathbf{A}) \cdot \frac{\partial^k \mathbf{C}}{\partial y^k_{(j)}} = \frac{\partial^k \mathbf{F}}{\partial y^k_{(j)}}, \quad j = 1 ... \nu \tag{3.13}$$

In equation (3.14), $\frac{\partial^{k-i}(\mathbf{A}^T\mathbf{A})}{\partial y^{k-i}_{(j)}}$ is indeed continuous, since $k \leq n$ and $W \in C^n(\Omega)$. The $\frac{\partial^i \mathbf{C}}{\partial y^i_{(j)}}$ are assumed to be continuous. $\mathbf{A}^T\mathbf{A}$ itself is continuous because W^n is continuous, and $\mathbf{A}^T\mathbf{A}$ is invertible by assumption. Thus $\frac{\partial^k \mathbf{C}}{\partial y^k_{(j)}}$ must also be continuous for any $k \leq n$. This concludes the proof.

3.2. Practical Computation of $\Pi^n_{\omega_N} f$

So far, we exhibited some theoretical considerations about the newly established smoothed particle operator. In the present section we would like to consider how to compute a smoothed particle approximation of a function f in practice with the method introduced in this chapter.

Again, for any arbitrary point $\mathbf{y}$, we have to solve the minimization problem

$$D = \sum_{i=1}^{N} \mid W_i \, (\, f_i - p(\mathbf{y}, \mathbf{x}_i) \,) \mid^2$$

where $W_i := W^n(\, r(\mathbf{y}, \mathbf{x}_i) \,) \, m_i$, f_i are the discrete function values, and moreover $p(\mathbf{y}, .) \in \mathcal{P}^d$ which means it is of the form

$$p(\mathbf{y}, \mathbf{x}) = \sum_{|\alpha| \leq d} C_\alpha(\mathbf{y}) \, (\mathbf{y} - \mathbf{x})^\alpha$$

In other words, we are looking for the constants $C_\alpha(\mathbf{y})$ such that the functional D is minimized.

As usual, the problem in matrix vector form is

$$\| \mathbf{A} \cdot \mathbf{C} - \mathbf{F} \|_2 \overset{!}{=} min$$

as it was shown in this chapter. The solution to this problem is given by

$$\mathbf{C} = \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \cdot (\mathbf{A}^T \mathbf{F}) \tag{3.14}$$

$\mathbf{C}$ itself contains the value of $\Pi^n_{\omega_N} f(\mathbf{y})$, because $\Pi^n_{\omega_N} f(\mathbf{y}) = p(\mathbf{y}, \mathbf{y}) = C_{\alpha_0}(\mathbf{y})$. So, with the solution of $\mathbf{C}$ we obtain immediately the value of $\Pi^n_{\omega_N} f(\mathbf{y})$.

We are now going to exhibit the matrices $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{F}$ for the most important cases $d = 0$ and $d = 1$.

1.) For the case $d = 0$, $\mathbf{A}^T\mathbf{A}$ is simply a one-by-one-matrix and $\mathbf{A}^T\mathbf{F}$ is a vector of length one. They have the form

$$\mathbf{A}^T\mathbf{A} = \left[\sum_{i=1}^{N} W_i^2 \right] \tag{3.15}$$

and

$$\mathbf{A}^T\mathbf{F} = \left[\sum_{i=1}^{N} W_i^2 \, f_i \right] \tag{3.16}$$

If we look a little closer, then we realize that $d = 0$ represents the classical smoothed particle approach with re-normalization, because

$$\Pi^n_{\omega_N} f(\mathbf{y}) = C_{\alpha_0}(\mathbf{y}) = \mathbf{C}(\mathbf{y}) = \frac{1}{\sum_{i=1}^{N} W_i^2} \sum_{i=1}^{N} W_i^2 f_i$$

The term $\sum\limits_{i=1}^{N} W_i^2 f_i$ is, in some sense, the classical smoothed particle approach, and $\frac{1}{\sum_{i=1}^{N} W_i^2}$ is a re-normalizing factor.

2.) For the case $d = 1$ we have

$$\mathbf{A}^T\mathbf{A} = \sum_{i=1}^{N} W_i^2 \begin{bmatrix} 1 & \xi_i^{(1)} & \xi_i^{(2)} & \xi_i^{(3)} \\ \xi_i^{(1)} & \xi_i^{(1)}\xi_i^{(1)} & \xi_i^{(2)}\xi_i^{(1)} & \xi_i^{(3)}\xi_i^{(1)} \\ \xi_i^{(2)} & \xi_i^{(1)}\xi_i^{(2)} & \xi_i^{(2)}\xi_i^{(2)} & \xi_i^{(3)}\xi_i^{(2)} \\ \xi_i^{(3)} & \xi_i^{(1)}\xi_i^{(3)} & \xi_i^{(2)}\xi_i^{(3)} & \xi_i^{(3)}\xi_i^{(3)} \end{bmatrix} \tag{3.17}$$

$$\mathbf{A}^T\mathbf{F} = \sum_{i=1}^{N} W_i^2 \begin{bmatrix} f_i \\ \xi_i^{(1)} f_i \\ \xi_i^{(2)} f_i \\ \xi_i^{(3)} f_i \end{bmatrix} \tag{3.18}$$

where $\xi_i^{(j)} := (y^{(j)} - x_i^{(j)})$ for $j \in \{1, ..., \nu\}$. The case $d = 1$ has no relation to the classical SPH any more. It is a very accurate method to compute a smoothed particle approximation $\Pi_{\omega_N}^n f$. For $d = 1$ in three dimensions, the computation of $\Pi_{\omega_N}^n f$ is about six times more expansive than in classical SPH. However, in comparison to classical SPH we are able to choose much larger smoothing lengths h in order to obtain a comparable accuracy. So, with larger h we also may choose larger time step sizes Δt which compensates the loss of computation speed.

The practical computation of $\Pi_{\omega_N}^n f$ now consists in

- establishing the matrix $\mathbf{A}^T\mathbf{A}$ and the vector $\mathbf{A}^T\mathbf{F}$
- solving the system (3.14)

These are now very simple operations.

We have to say some words about why solving the least squares problem using $(\mathbf{A}^T\mathbf{A})^{-1}$ instead of using some Q-R-decomposition of the matrix $\mathbf{A}$. There are two reasons.

- computing $(\mathbf{A}^T\mathbf{A})$ and solving the appropriate system is much faster than solving it by a Q-R-decomposition because $(\mathbf{A}^T\mathbf{A})$ degenerates to a very small matrix (in the case $d = 0$ it even reduces to a one-by-one matrix)
- we take advantage of the formulation with $(\mathbf{A}^T\mathbf{A})$ in the next section where we are going to practically compute $\nabla\Pi_{\omega_N}^n f$.

3.3. Practical Computation of $\nabla\Pi_{\omega_N}^n f$

In section 3.2 we showed how to compute a smoothed particle approximation $\Pi_{\omega_N}^n f$ of an arbitrary function f. However, we remember that $\Pi_{\omega_N}^n f$ was originally established in order to compute approximations of the gradient of some function f. Therefore, we are not very much interested in computing $\Pi_{\omega_N}^n f$, but we would like to compute $\nabla\Pi_{\omega_N}^n f$.

Suppose, we would like to compute the spatial derivative $\frac{\partial}{\partial y^{(j)}} \Pi^n_{\omega_N} f(\mathbf{y})$ for arbitrary $\mathbf{y} \in \Omega(\bar{t})$. With

$$\Pi^n_{\omega_N} f(\mathbf{y}) = C_{\alpha_0}(\mathbf{y})$$

we indeed obtain

$$\frac{\partial}{\partial y^{(j)}} \Pi^n_{\omega_N} f(\mathbf{y}) = \frac{\partial}{\partial y^{(j)}} C_{\alpha_0}(\mathbf{y})$$

and so, we have to compute simply $\frac{\partial \mathbf{C}}{\partial y^{(j)}}$ in order to obtain $\frac{\partial}{\partial y^{(j)}} \Pi^n_{\omega_N} f(\mathbf{y})$. The question arises, how to do it. We rewrite equation (3.14) by

$$\left(\mathbf{A}^T \mathbf{A} \right) \cdot \mathbf{C} = \mathbf{A}^T \mathbf{F} \tag{3.19}$$

We now perform the differentiation $\frac{\partial}{\partial y^{(j)}}$ of equation (3.19)

$$\frac{\partial \left(\mathbf{A}^T \mathbf{A} \right)}{\partial y^{(j)}} \cdot \mathbf{C} \; + \; \left(\mathbf{A}^T \mathbf{A} \right) \cdot \frac{\partial \mathbf{C}}{\partial y^{(j)}} \; = \; \frac{\partial (\mathbf{A}^T \mathbf{F})}{\partial y^{(j)}} \tag{3.20}$$

Finally, (3.20) gives

$$\frac{\partial \mathbf{C}}{\partial y^{(j)}} \; = \; \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \cdot \left(\frac{\partial (\mathbf{A}^T \mathbf{F})}{\partial y^{(j)}} \; - \; \frac{\partial \left(\mathbf{A}^T \mathbf{A} \right)}{\partial y^{(j)}} \cdot \mathbf{C} \right) \tag{3.21}$$

In addition to the steps necessary for computing $\Pi^n_{\omega_N} f(\mathbf{y})$ we have to establish the derivatives of the matrix $\mathbf{A}^T \mathbf{A}$ and of the vector $\mathbf{A}^T \mathbf{F}$. This may be done analytically as a brief look at equations (3.15), (3.16) for $d = 0$ or at (3.17), (3.18) for $d = 1$ tells. In general, we can say that an entry of the matrix $\mathbf{A}^T \, \mathbf{A}$ has the form

$$a \;\; = \;\; \sum_{i=1}^{N} (W_i)^2 \; (\mathbf{y} - \mathbf{x}_i)^{\alpha}$$

Thus, the derivative of such an entry is to be computed analytically by

$$\frac{\partial a}{\partial y^{(j)}} \;\; = \;\; \frac{4}{h_i^2} \sum_{i=1}^{N} W_i \; (W_i)' \; (\mathbf{y} - \mathbf{x}_i)^{\alpha + \mathbf{e}_j} \; + \; \sum_{i=1}^{N} (W_i)^2 \; \alpha^{(j)} \, (\mathbf{y} - \mathbf{x}_i)^{\alpha - \mathbf{e}_j}$$

We have the abbreviations

$$\begin{aligned} W_i \;\; &= \;\; W^n(\; r(\mathbf{y}, \mathbf{x}_i) \;)\, m_i \\ (W_i)' \;\; &= \;\; \frac{dW^n}{dr}(\; r(\mathbf{y}, \mathbf{x}_i) \;)\, m_i \end{aligned}$$

For computing the spatial derivatives of $\mathbf{A}^T\,\mathbf{F}$, we realize that an entry of this vector looks like

$$b \;=\; \sum_{i=1}^{N} (W_i)^2\;(\mathbf{y}-\mathbf{x}_i)^{\alpha}\cdot f_i$$

The computation of spatial derivatives of b is as easy as for a.

3.4. Practical Computation of Second Derivatives of $\Pi^n_{\omega_N} f(\mathbf{y})$

Whenever we would like to study Navier-Stokes flow and/or heat-transfer problems, then the computation of second spatial derivatives is necessary. In this section we will emphasize that for $d > 0$ we are able to compute smoothed approximations of second derivatives of functions in a direct way. This idea is based on the following fact. We know, that the vector $(C_{\mathbf{e}_1}, ..., C_{\mathbf{e}_\nu})$ (which is computed anyway if $d > 0$) is an approximation of $-\nabla f$. $-(C_{\mathbf{e}_1}, ..., C_{\mathbf{e}_\nu})$ does not exactly assume the same value as $\nabla\Pi^n_{\omega_N} f$, but the two are close. The only difference between them is, that $\nabla\Pi^n_{\omega_N} f$ is the gradient of an existing smooth function, but $-(C_{\mathbf{e}_1}, ..., C_{\mathbf{e}_\nu})$ is not necessarily gradient to some function. However, in order to compute approximations of the second derivative of f, we use the derivatives

$$-\frac{\partial}{\partial y^{(j)}}(C_{\mathbf{e}_1}, ..., C_{\mathbf{e}_\nu}), \quad j \in \{1; ...; \nu\}$$

In this sense, for example

$$-\frac{\partial C_{\mathbf{e}_1}}{\partial y^{(1)}}$$

is an approximation of $\frac{\partial^2 f}{\partial y^{(1)} \partial y^{(1)}}$, or, in the same way,

$$-\frac{\partial C_{\mathbf{e}_2}}{\partial y^{(1)}}$$

is an approximation of $\frac{\partial^2 f}{\partial y^{(1)} \partial y^{(2)}}$. Thus, almost for free, we obtain approximations of second derivatives of some function f.

If d is the order of the polynomial, then $C_{\mathbf{e}_i}$ as an approximation of the partial derivative with respect to $y^{(i)}$ of the order $\mathbf{O}(h^{d+1})$. The proof of this statement is similar to the proof of theorem 3.3. However, this means that any spatial derivative of $C_{\mathbf{e}_i}$ is at least of order $\mathbf{O}(h^d)$ which is quite a convenient approximation property. Hence, for $d \geq 1$ the approximation of the second derivative of some function has a reasonable quality.

Chapter 4

4. Handling Boundary Conditions in the General SPH Setup

In this chapter we turn to incorporating boundary conditions into the particle scheme. Unlike in astrophysics, in fluid mechanics we have to deal with various types of boundaries, for example solid walls as well as inflow or outflow boundaries or even free surfaces. For boundary treatment in the general SPH scheme, we intend to consider the movement of characteristic curves and, consequently, we will have to obey the information which is carried along the characteristic curves. Please refer to the book of C. Hirsch (see [12]) where the characteristic behavior of the Euler equations is explained in detail. For any boundary, a characteristic curve tells us if a certain kind of information comes from "outside" towards $\partial\Omega$ or from "inside". For numerical boundary treatment, information along characteristic curves have to be taken into account only if they reach the boundary from "inside", that means if they carry information from the interior of the flow domain towards the boundary. If characteristic information is carried from the exterior to Ω (meaning the characteristic curves come from "outside"), then this characteristic information must not be involved into the particle setup, but it has to be replaced by some free boundary condition. In the first section (section 4.1) of this chapter, we will have a brief look at the characteristic form of the Euler equations. In the second section we are then going to incorporate the reviewed theory into the general particle scheme.

4.1. Characteristic Theory for Euler equations - Theoretical Considerations on Boundary Behavior

The theory given in this section is to be found in detail in [12] and is redrawn here briefly to cover all necessary details for treating boundaries in general SPH.

We would like to consider the 3D Euler equations. Unlike in chapter 2, where we introduced the Euler equations in conservative form, here, for convenience, we will use the Euler equations in primitive variable form. In this form, the Euler equations are

$$\mathbf{I} \cdot \frac{\partial \mathbf{V}}{\partial t} + \mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial x^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial x^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial x^{(3)}} = S \tag{4.1}$$

where

$$\mathbf{V} = (\rho, v^{(1)}, v^{(2)}, v^{(3)}, p)^T$$

$$S = \text{source terms}$$

$$\mathbf{A} = \begin{pmatrix} v^{(1)} & \rho & 0 & 0 & 0 \\ 0 & v^{(1)} & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & v^{(1)} & 0 & 0 \\ 0 & 0 & 0 & v^{(1)} & 0 \\ 0 & \rho c^2 & 0 & 0 & v^{(1)} \end{pmatrix} \mathbf{B} = \begin{pmatrix} v^{(2)} & 0 & \rho & 0 & 0 \\ 0 & v^{(2)} & 0 & 0 & 0 \\ 0 & 0 & v^{(2)} & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & v^{(2)} & 0 \\ 0 & 0 & \rho c^2 & 0 & v^{(2)} \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} v^{(3)} & 0 & 0 & \rho & 0 \\ 0 & v^{(3)} & 0 & 0 & 0 \\ 0 & 0 & v^{(3)} & 0 & 0 \\ 0 & 0 & 0 & v^{(3)} & \frac{1}{\rho} \\ 0 & 0 & 0 & \rho c^2 & v^{(3)} \end{pmatrix}$$

c denotes the speed of sound given by

$$c^2 = \left.\frac{\partial p}{\partial \rho}\right|_s$$

which is the partial derivative of the pressure with respect to the density ρ under the condition of constant entropy. Usually, p is given as a function of ρ and T. Hence, upon the knowledge of $\frac{\partial p}{\partial \rho}$ and $\frac{\partial p}{\partial T}$, we can rewrite

$$c^2 = \frac{\partial p}{\partial \rho} + \frac{p}{c_v\, \rho^2} \frac{\partial p}{\partial T}$$

Now, we would like to consider the spreading of information at some point $\mathbf{x} \in \Omega(t)$ in a certain direction represented by the normal vector $\mathbf{n} := (n^{(1)}, n^{(2)}, n^{(3)})$ with $\|\mathbf{n}\|_2 = 1$. In order to achieve this, we establish surfaces $s(t, \mathbf{x}) = 0$, $\mathbf{x} := (x^{(1)}, x^{(2)}, x^{(3)})$. We require, that these surfaces be perpendicular to the normal direction $\mathbf{n}$ and that $\|\nabla s\| = 1$, that is

$$\frac{\partial s}{\partial x^{(i)}} = n^{(i)} \ , \quad i = 1, ..., 3 \tag{4.2}$$

We assume furthermore, that the surface travels with a certain velocity $\mathbf{v}_s$. This velocity, naturally, points into the direction of $\mathbf{n}$ since $\mathbf{n}$ is perpendicular to the surface. Hence we have

$$\mathbf{v}_s = v_s \cdot \mathbf{n} \tag{4.3}$$

where $|v_s|$ is the magnitude of the velocity. If we take the total time derivative of s with respect to $\mathbf{v}_s$, then we obtain

$$\begin{aligned} \frac{ds}{dt} &= 0 \Rightarrow && (4.4) \\ \frac{\partial s}{\partial t} + v_s^{(1)} \frac{\partial s}{\partial x^{(1)}} + v_s^{(2)} \frac{\partial s}{\partial x^{(2)}} + v_s^{(3)} \frac{\partial s}{\partial x^{(3)}} &= 0 \Rightarrow && (4.5) \end{aligned}$$

With $v_s^{(i)} = v_s \cdot n^{(i)}$ and $\frac{\partial s}{\partial x^{(i)}} = n^{(i)}$ we have

$$\begin{aligned} \frac{\partial s}{\partial t} + v_s \left(\left(n^{(1)}\right)^2 + \left(n^{(2)}\right)^2 + \left(n^{(3)}\right)^2 \right) &= 0 \Rightarrow && (4.6) \\ \frac{\partial s}{\partial t} + v_s &= 0 \Rightarrow && (4.7) \\ \frac{\partial s}{\partial t} &= -v_s && (4.8) \end{aligned}$$

This means that the partial time derivative of s coincides with the negative speed of the surface viewed in normal direction. Up to now, the surface s has nothing in special. For finding the characteristic form of equation (4.1), we introduce another definition.

Definition 4.1.: (Characteristic Surfaces)
We call a surface $s(t, \mathbf{x}) = 0$ with $\|\nabla s\|_2 = 1$ a characteristic surface to the hyperbolic equation (4.1), if it meets the requirement

$$\mathbf{l} \cdot \left(\mathbf{I} \cdot \frac{\partial s}{\partial t} + \mathbf{A} \cdot \frac{\partial s}{\partial x^{(1)}} + \mathbf{B} \cdot \frac{\partial s}{\partial x^{(2)}} + \mathbf{C} \cdot \frac{\partial s}{\partial x^{(3)}} \right) = \mathbf{0} \tag{4.9}$$

for some vector $\mathbf{l}$.
Remark:

- $-\frac{\partial s}{\partial t}$ is the eigenvalue of the matrix $\left(\mathbf{A} \cdot \frac{\partial s}{\partial x^{(1)}} + \mathbf{B} \cdot \frac{\partial s}{\partial x^{(2)}} + \mathbf{C} \cdot \frac{\partial s}{\partial x^{(3)}}\right)$
- $\mathbf{l}$ is left eigenvector of the matrix $\left(\mathbf{A} \cdot \frac{\partial s}{\partial x^{(1)}} + \mathbf{B} \cdot \frac{\partial s}{\partial x^{(2)}} + \mathbf{C} \cdot \frac{\partial s}{\partial x^{(3)}}\right)$

Definition 4.1 seems to make no sense, there is no reason yet why to define a characteristic surface like this. In the following, we will show where the definition above finds its roots and that it indeed makes some sense by helping us to write down a characteristic form of equation (4.1).

In the following we sketch the steps of establishing the characteristic form of equation (4.1).

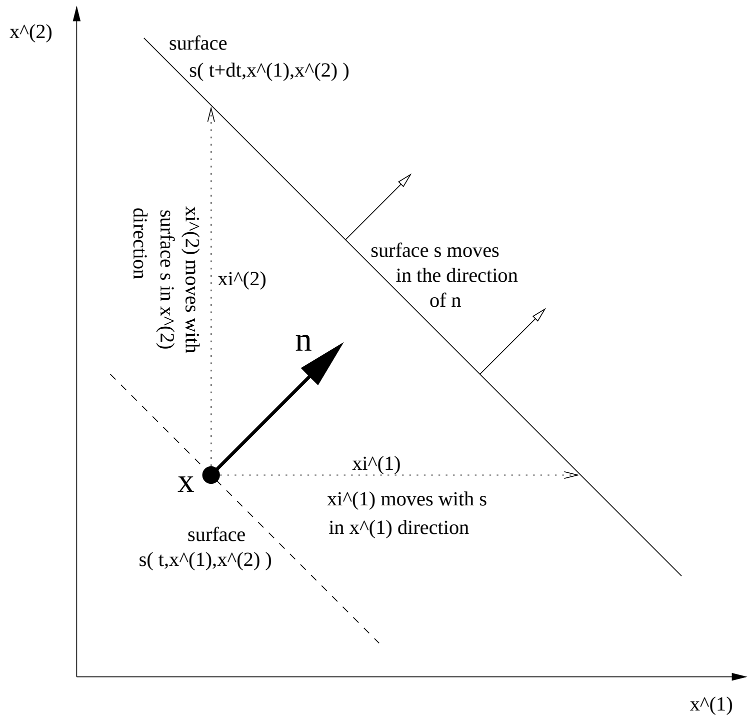
first step:
Let some direction vector $\mathbf{n}$ be given. $\mathbf{n}$ is the direction into which we would like to trace information. With the help of some not yet known, but fixed surface $s(t, \mathbf{x}) = 0$ being perpendicular to $\mathbf{n}$ with $\|\nabla s\|_2 = 1$ (remember that in this case we have $\nabla s = \mathbf{n}$ and $v_s = -\frac{\partial s}{\partial t}$), we introduce a coordinate transformation $(t, x^{(1)}, x^{(2)}, x^{(3)}) \to (\tau, \xi^{(1)}, \xi^{(2)}, \xi^{(3)})$ as follows.

$$\begin{aligned} \tau &= t - \frac{n^{(1)}}{v_s} x^{(1)} - \frac{n^{(2)}}{v_s} x^{(2)} - \frac{n^{(3)}}{v_s} x^{(3)} && (4.10) \\ \xi^{(i)} &= x^{(i)}\,, \quad i = 1, 2, 3 && (4.11) \\ & && (4.12) \end{aligned}$$

This leads to the following differential operators

$$\begin{aligned} \frac{\partial}{\partial t} &= \frac{\partial}{\partial \tau} && (4.13) \\ \frac{\partial}{\partial x^{(i)}} &= -\frac{n^{(i)}}{v_s} \cdot \frac{\partial}{\partial \tau} + \frac{\partial}{\partial \xi^{(i)}} && (4.14) \end{aligned}$$

Remark: This formulas coincide with the ones given in [12] in section 3.3, equations (3.3.4). We would like to concentrate the attention to the derivatives $\frac{\partial}{\partial \xi^{(i)}}$. The value $\frac{\partial f}{\partial \xi^{(i)}}$ of some smooth function $f : \bar{\Omega} \to \mathbf{R}$ represents exactly the change of f one would experience by moving with the surface s in the direction of $x^{(i)}$, i.e. $\xi^{(i)}$ is the variable that moves with the surface s in the direction of $x^{(i)}$. The following plot tries to give a geometrical understanding of these derivatives.


x^(2)
surface
s(t+dt,x^(1),x^(2))
xi^(2) moves with surface s in x^(2) direction
xi^(2)
surface s moves in the direction of n
n
x
xi^(1)
xi^(1) moves with s in x^(1) direction
surface
s(t,x^(1),x^(2))
x^(1)

$\frac{\partial}{\partial \xi^{(i)}}$ are therefore called derivatives along the surface s.

second step:
By the help of these newly developed differential operators, we are able to rewrite equation (4.1). We simply replace the derivatives in (4.1) by the expressions obtained in (4.13/4.14). This gives

$$\begin{aligned} &-\left(-v_s \cdot \mathbf{I} + n^{(1)} \cdot \mathbf{A} + n^{(2)} \cdot \mathbf{B} + n^{(3)} \cdot \mathbf{C} \right) \cdot \frac{\partial \mathbf{V}}{\partial \tau} + \\ &\qquad v_s \cdot \left(\mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(3)}} \right) \; = \; v_s \cdot S \end{aligned} \tag{4.15}$$

Finally we multiply equation (4.15) from left by a vector **l**, which gives

$$\begin{aligned} &-\mathbf{l} \cdot \left(-v_s \cdot \mathbf{I} + n^{(1)} \cdot \mathbf{A} + n^{(2)} \cdot \mathbf{B} + n^{(3)} \cdot \mathbf{C} \right) \cdot \frac{\partial \mathbf{V}}{\partial \tau} + \\ &\qquad v_s \cdot \mathbf{l} \cdot \left(\mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(3)}} \right) \; = \; v_s \cdot \mathbf{l} \cdot S \end{aligned} \tag{4.16}$$

or equivalently

$$\begin{aligned} &-\mathbf{l} \cdot \left(\frac{\partial s}{\partial t} \cdot \mathbf{I} + \frac{\partial s}{\partial x^{(1)}} \cdot \mathbf{A} + \frac{\partial s}{\partial x^{(2)}} \cdot \mathbf{B} + \frac{\partial s}{\partial x^{(3)}} \cdot \mathbf{C} \right) \cdot \frac{\partial \mathbf{V}}{\partial \tau} + \\ &\qquad v_s \cdot \mathbf{l} \cdot \left(\mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(3)}} \right) \; = \; v_s \cdot \mathbf{l} \cdot S \end{aligned}$$

third step:

We immediately realize that equation (4.16) reduces to

$$\mathbf{l} \cdot \left(\mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(3)}} \right) = \mathbf{l} \cdot S \tag{4.17}$$

if s is a characteristic surface and $\mathbf{l}$ is an eigenvector corresponding to the eigenvalue $\lambda := -\frac{\partial s}{\partial t} = v_s$. We observe that only derivatives along a characteristic surface are relevant if one moves along the characteristic surface. For 3D Euler equations, we have five eigenvalues and also, as shown later, five linearly independent eigenvectors, hence the equivalent writing to equation (4.1) in characteristic form is

$$\mathbf{l}_i \cdot \left(\mathbf{A} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(1)}} + \mathbf{B} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(2)}} + \mathbf{C} \cdot \frac{\partial \mathbf{V}}{\partial \xi^{(3)}} \right) \;=\; \mathbf{l}_i \cdot S \qquad i = 1...5 \tag{4.18}$$

where $\mathbf{l}_i$ denotes the left eigenvector which corresponds to the i-th eigenvalue $\lambda_i := -\frac{\partial s_i}{\partial t} = (v_s)_i$. Also, we have five different characteristic surfaces denoted by $s_i(t, \mathbf{x})$. Equations (4.18) are called compatibility relations (see [12], section 3.3.1) or also characteristic information.

Now, when a characteristic surface enters the flow domain from outside, then naturally characteristic information is carried towards the boundary from outside. In the case of computational fluid flow, information does not exist outside of Ω in the form of computational data. Hence, the characteristic information being carried from outside have to be replaced by free boundary conditions $bc(t, \mathbf{V}) = 0, \quad bc : T \times \mathbf{R}^5 \rightarrow \mathbf{R}$.

This shows that for solving fluid flow, at the boundaries it is important to know, where characteristic information comes from in order to replace it, if necessary, by free boundary conditions. Let $\mathbf{n}$ be the inward pointing normal to the boundary at some point $\mathbf{y} \in \partial\Omega$, and let s_i be the characteristic surface moving into the direction of $\mathbf{n}$ which corresponds to the i-th eigenvalue λ_i, then the sign of $(v_s)_i := \lambda_i$ gives us the direction of the propagation of the information. That is, if the characteristic speed is positive, (i.e. if λ_i is positive) then s_i obviously enters the flow domain Ω and hence, for this surface, information has to be set in form of boundary conditions. Therefore, in the following we will exhibit the eigenvalues λ_i of the set of Euler equations (4.1) and the corresponding left eigenvectors $\mathbf{l}_i$ in order to decide, what characteristic surface enters Ω and what does not.

The eigenvalues in the Euler case are

$$\begin{array}{rcl} \lambda_1 & = & (\mathbf{v} \cdot \mathbf{n}) \\ \lambda_2 & = & (\mathbf{v} \cdot \mathbf{n}) \\ \lambda_3 & = & (\mathbf{v} \cdot \mathbf{n}) \\ \lambda_4 & = & (\mathbf{v} \cdot \mathbf{n} - c) \\ \lambda_5 & = & (\mathbf{v} \cdot \mathbf{n} + c) \end{array} \tag{4.19}$$

The corresponding eigenvectors are

$$\begin{array}{lcccccccl}
\mathbf{l}_1 & = & (& c^2 & 0 & \rho c n^{(3)} & -\rho c n^{(2)} & -1 &) \\
\mathbf{l}_2 & = & (& c^2 & -\rho c n^{(3)} & 0 & \rho c n^{(1)} & -1 &) \\
\mathbf{l}_3 & = & (& c^2 & \rho c n^{(2)} & -\rho c n^{(1)} & 0 & -1 &) \\
\mathbf{l}_4 & = & (& 0 & \rho c n^{(1)} & \rho c n^{(2)} & \rho c n^{(3)} & -1 &) \\
\mathbf{l}_5 & = & (& 0 & \rho c n^{(1)} & \rho c n^{(2)} & \rho c n^{(3)} & 1 &)
\end{array} \tag{4.20}$$

With the eigenvalues and eigenvectors, we obtain the five compatibility relations

$$\begin{aligned}
c^2\frac{d\rho}{dt} + \rho c n^{(3)}\frac{du^{(2)}}{dt} - \rho c n^{(2)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(2)}\frac{\partial p}{\partial x^{(3)}} - c n^{(3)}\frac{\partial p}{\partial x^{(2)}} + \mathbf{l}_1 \cdot S \\
c^2\frac{d\rho}{dt} - \rho c n^{(3)}\frac{dv^{(1)}}{dt} + \rho c n^{(1)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(3)}\frac{\partial p}{\partial x^{(1)}} - c n^{(1)}\frac{\partial p}{\partial x^{(3)}} + \mathbf{l}_2 \cdot S \\
c^2\frac{d\rho}{dt} + \rho c n^{(2)}\frac{dv^{(1)}}{dt} - \rho c n^{(1)}\frac{dv^{(2)}}{dt} - \frac{dp}{dt} &= c n^{(1)}\frac{\partial p}{\partial x^{(2)}} - c n^{(2)}\frac{\partial p}{\partial x^{(1)}} + \mathbf{l}_3 \cdot S \\
\rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= \rho c^2 \cdot \nabla \mathbf{v} - c(\mathbf{n} \cdot \nabla p) + \mathbf{l}_4 \cdot S \\
\rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} + \frac{dp}{dt} &= -\rho c^2 \cdot \nabla \mathbf{v} - c(\mathbf{n} \cdot \nabla p) + \mathbf{l}_5 \cdot S
\end{aligned} \tag{4.21}$$

which might be rewritten just in matrix form as

$$\mathbf{L} \cdot \frac{d\mathbf{V}}{dt} = \mathbf{r}\,; \quad \mathbf{L} \in \mathbf{R}^{5\times 5}\,; \quad \mathbf{r} \in \mathbf{R}^5 \tag{4.22}$$

We would like to emphasize, that this form is another form of writing down the Euler equations which is completely equivalent to equations (4.1). It is rewritten for the purpose of tracing information.

With the correspondence between primitive and conservative variables, given by

$$\frac{d\Phi}{dt} = \mathbf{M} \cdot \frac{d\mathbf{V}}{dt} \tag{4.23}$$

where

$$\mathbf{M} := \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
v^{(1)} & \rho & 0 & 0 & 0 \\
v^{(2)} & 0 & \rho & 0 & 0 \\
v^{(3)} & 0 & 0 & \rho & 0 \\
E + \rho\frac{\partial u}{\partial \rho} & \rho v^{(1)} & \rho v^{(2)} & \rho v^{(3)} & \rho\frac{\partial u}{\partial p}
\end{pmatrix}$$

we can rewrite (4.22) as

$$\mathcal{L} \cdot \frac{d\Phi}{dt} = \mathcal{R} \tag{4.24}$$

where
$\mathcal{L} := \mathbf{L} \cdot \mathbf{M}^{-1}$ and
$\mathcal{R} := \mathbf{r}$

We realize that

$$\mathcal{L}^{(i,.)} \cdot \frac{d\Phi}{dt} = \mathcal{R}^{(i)} \; ; \;\; i = 1, ..., N$$

denotes the i-th compatibility relation written in conservative variable form.

We have to make clear that the right hand side of (4.24) (vector $\mathcal{R}$) depends on $\nabla \mathbf{v}$ and ∇p. So it is obvious, that instead of

$$\mathcal{L} \cdot \frac{d\Phi}{dt} = \mathcal{R}(\nabla \mathbf{v}, \nabla p)$$

we have to write

$$\mathcal{L} \cdot \frac{d\Phi}{dt} = \mathcal{R}(\nabla \Pi^n_{\omega_N} \mathbf{v}, \nabla \Pi^n_{\omega_N} p) \tag{4.25}$$

using the smoothed particle approximations instead of the exact derivatives.

Up to now, we have written down compatibility relations in conservative variable form. These represent pieces of information that propagate with a certain speed along a certain characteristic surface. For treating boundaries, we will now have to pick out those compatibility relations which are relevant (i.e. which propagate on a characteristic surface coming from inside of the flow domain towards the boundary).

For the next sections, we denote by $\mathbf{n}$ the inward pointing normal vector to some point of $\partial\Omega$.

4.1.1. Inflow Boundaries

At inflow boundaries, clearly we have

$$\mathbf{v} \cdot \mathbf{n} > 0$$

This means that s_1, s_2, s_3 and s_5 enter the flow domain. If

$$\mathbf{v} \cdot \mathbf{n} > c \quad \text{supersonic inflow}$$

then also s_4 enters the flow domain. Hence, at inflow boundaries with subsonic inflow, we can involve only compatibility relation 4 into our computation. We have to set four other conditions bc_1, bc_2, bc_3, bc_4 to be involved in the computations.

For supersonic inflow, we are not allowed to use any of the given compatibility relations. Thus, we are required to set five independent boundary conditions bc_1, bc_2, bc_3, bc_4, bc_5.

4.1.2. Outflow Boundaries

At outflow boundaries, clearly we have

$$\mathbf{v} \cdot \mathbf{n} < 0$$

If

$$\mathbf{v} \cdot \mathbf{n} > -c \quad \text{subsonic outflow}$$

then only characteristic surface s_5 enters the flow domain, hence we are required to set bc_5, but we are allowed to use compatibility relations 1, 2, 3 and 4. If

$$\mathbf{v} \cdot \mathbf{n} < -c \quad \text{supersonic outflow}$$

then we are requested to use all of the compatibility relations, hence no user set boundary condition is required.

4.1.3. Solid Wall Boundaries

For solid walls, we have

$$\mathbf{v} \cdot \mathbf{n} = 0$$

thus only characteristic surface s_5 enters the flow domain, and consequently compatibility relations 1, 2, 3, 4 may be used, and we have to set bc_5 by

$$bc_5 = \mathbf{v} \cdot \mathbf{n} = 0$$

4.2. Practical Setting of Boundary Conditions

4.2.1. Boundary Treatment for Euler Equations

Our technique to set boundary treatment into praxis is to place boundary particles along the boundaries. These boundary particles have all the properties which a fluid particle would have, i.e boundary particle j carries information Φ_j and $\mathbf{x}_j$. In addition to that, it carries also the information $\mathbf{n}_j$, the inward pointing normal to the boundary. The difference to usual fluid particles is that boundary particles do not move essentially with the velocity of the fluid flow since we require that they be stuck to the geometrical boundary. Hence, for boundary particle j we have

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{v}_{p_j}$$

where $\mathbf{v}_{p_j} := \mathbf{v}_p(\mathbf{x}_i)$ denotes the velocity which boundary particle j is physically moving with.

In the previous section we have seen, that for any kind of boundary we have a set of compatibility relations which may be written as

$$\mathcal{L}^{(i,.)} \cdot \frac{d\Phi}{dt} = \mathcal{R}^{(i)} \quad i \in I_{cr} \tag{4.26}$$

where the set I_{cr} contains all the numbers of those compatibility relations which correspond to characteristic surfaces coming from inside towards the boundary. Since the total time derivative $\frac{d}{dt}$ corresponds to the fluid velocity $\Pi\mathbf{v}$ we have to rewrite the total time derivative such that it corresponds to the velocity $\mathbf{v}_p$ which the boundary particle actually moves with, that is

$$\frac{d_{\mathbf{v}_p}}{dt} = \frac{d}{dt} + (\mathbf{v}_p - \Pi^n_{\omega_N}\mathbf{v}) \cdot \nabla \tag{4.27}$$

Using this, for boundary particle j, equation (4.26) reads as

$$\mathcal{L}^{(i,.)} \cdot \frac{d_{\mathbf{v}_p}\Phi_j}{dt} \;=\; \mathcal{R}^{(i)} + \mathcal{L}^{(i,.)} \cdot (\mathbf{v}_p - \Pi^n_{\omega_N}\mathbf{v}) \cdot \nabla\Phi \;=:\; \bar{\mathcal{R}}^{(i)} \quad i \in I_{cr} \tag{4.28}$$

where the total derivative $\frac{d_{\mathbf{v}_p}}{dt}$ corresponds to the velocity $\mathbf{v}_p$ which the boundary particle is moving with.

As mentioned previously, the set of compatibility relations (4.28) is completed by the set of free boundary conditions

$$(bc_i)_j(t,\Phi) = 0 \quad i \in I_{bc} \tag{4.29}$$

where I_{bc} denotes the set $\{1,2,3,4,5\} - I_{cr}$ and j denotes the index of the boundary particle concerned.

These free boundary conditions are not globally, but only locally defined. $(bc_i)_j$ are valid only for the position of boundary particle j. Hence, for the local free boundary condition we have

$$\frac{\partial(bc_i)_j}{\partial t} + \frac{\partial(bc_i)_j}{\partial \Phi} \cdot \frac{d_{\mathbf{v}_p}\Phi_j}{dt} = 0 \qquad i \in I_{bc} \tag{4.30}$$

We can rewrite (4.30) in a more convenient form

$$L^{(i,\cdot)} \cdot \frac{d_{\mathbf{v}_p}\Phi_j}{dt} = R^{(i)} \qquad i \in I_{bc} \tag{4.31}$$

where obviously $L^{(i,\cdot)} := \frac{\partial(bc_i)_j}{\partial \Phi}$ is the i-th row in the matrix L and $R^{(i)} := -\frac{\partial(bc_i)_j}{\partial t}$ is the i-th component of vector R. With equation (4.28) and (4.31) we have a linear system completely describing the unknowns

$$\frac{d_{\mathbf{v}_p}\Phi_j}{dt}$$

for the boundary particle with index j. This system is easily to be solved and the changes of Φ_j are obtained. We have to make sure that the bc_i are chosen such that the solvability of the linear system (4.28/4.31) is not violated. If the system turns out to be unsolvable, then some free boundary condition is linearly dependent on some compatibility relation.

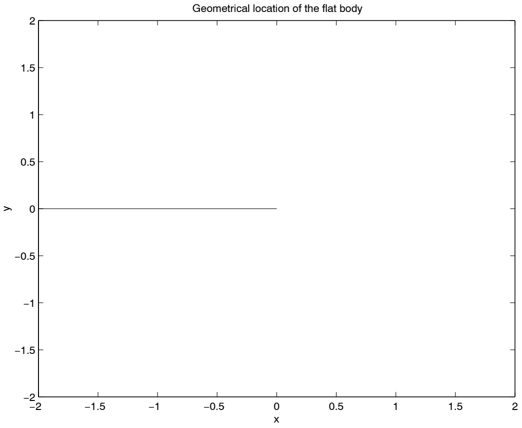
4.2.2. Boundary Treatment for Navier-Stokes Equations

The numerical setting of boundary conditions for the Navier-Stokes equations are similar to the one for the Euler equations. Unlike for Euler, where S contains only gravity terms, for boundary treatment concerning Navier-Stokes we are required to add the viscous terms to S in equation (4.1). Having done this, all technical steps are the same as for Euler.

4.3. Treatment of Sharp Edged Solid Wall Boundaries

Although the treatment of boundaries using the method as described above works very well for smooth boundaries, it causes some problems when a solid wall boundary proves to have sharp edges. The problem is sketched in the following pictures.

Suppose we consider a problem where a geometrically flat body (as for example a sheet of paper given by $\mathbf{FB} := \{ \mathbf{y} \mid y^{(1)} \leq 0 \;,\; y^{(2)} = 0 \}$, see sketch below) is placed in a fluid flow.



The sharp edge in our example is placed at the position $\mathbf{x}_c = (0,0)^T$. In chapter 3 we introduced the method how to establish a very accurate smoothed particle function $\Pi^n_{\omega_N} f$. For constructing a least squares approach by the help of a function $W^n(\ r(\mathbf{y},\mathbf{x})\ )$ giving weight to particles with index i situated around the position $\mathbf{y}$, we used a very simple distance function

$$r(\mathbf{y},\mathbf{x}) = \left(\frac{\|\mathbf{y}-\mathbf{x}\|_2}{h(\mathbf{x})} \right)^2$$

For sharp edges, this approach will not work. That is why for sharp edges a particle has to "look around the corner" a little bit, but the distance function above is based on the straight distance between two points. Although far away from the sharp edges of the flat body we have no problem drawing the contour lines of W^n (they are circle shaped and are simply truncated at the flat body), it is not really clear how the contour lines have to be shaped in the neighborhood of the "edges".

The question is how far should a particle at the position $\mathbf{y}$ close to $\mathbf{x}_c$ look around the sharp edge. In order to keep smoothness properties of the operator Π^n, we have to keep smoothness of $W^n(r(\mathbf{y},\mathbf{x}))$, i.e. the distance function $r(\mathbf{y},\mathbf{x})$ has to be smooth. That means the weighting function has to look around the edge smoothly. To achieve this, we suggest to generalize the distance function $r(\mathbf{y},\mathbf{x})$ taking into account the eventual appearance of sharp edges. For this reason we come up with the following two definitions.

Definition 4.2:
Suppose $\mathbf{X}_c$ is the set of all geometrical points belonging to the sharp edge of $\Omega(t)$.

Let $\mathbf{y} \in \Omega(t)$ and $\mathbf{x} \in \Omega(t)$. Let $\mathbf{x}_p(\tau), \quad 0 \leq \tau \leq 1$ be the shortest connecting path between $\mathbf{y}$ and $\mathbf{x}$ such that $\mathbf{x}_p(\tau) \in \Omega(t) \quad \forall \tau \in [0,1]$. We call $\mathbf{y}$ and $\mathbf{x}$ indirectly connected, if there is $\bar{\tau} \in [0,1]$ such that $\mathbf{x}_p(\bar{\tau}) \in \mathbf{X}_c$, where we define $\mathbf{x}_c := \mathbf{x}_p(\bar{\tau})$. Otherwise we call $\mathbf{y}$ and $\mathbf{x}$ directly connected.

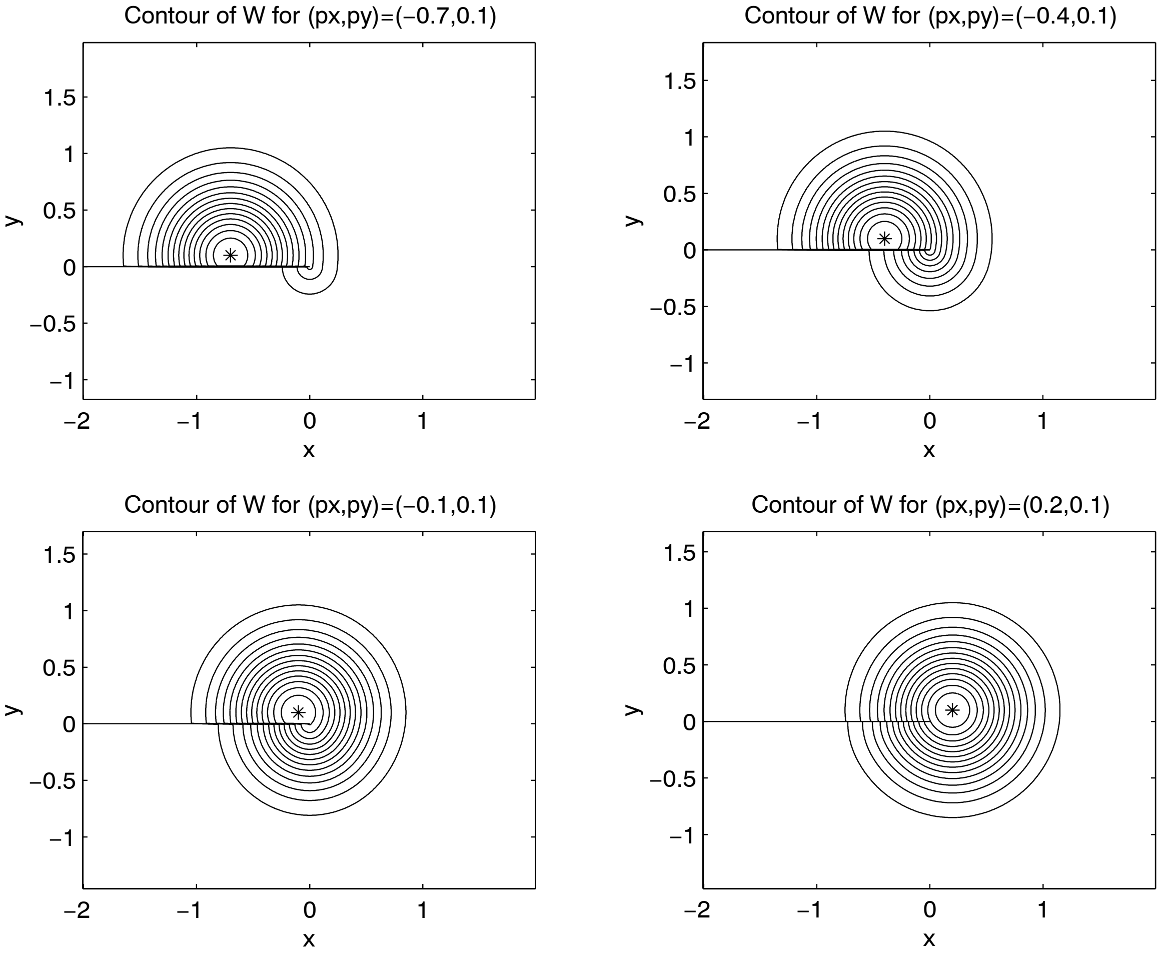
Definition 4.3. (Geodetic Distance Between Two Points)
Let $\mathbf{y} \in \Omega$ and $\mathbf{x} \in \Omega$. We define the distance between the two points by

$$r(\mathbf{y},\mathbf{x}) := \begin{cases} \left(\frac{\|\mathbf{y}-\mathbf{x}\|_2}{h(\mathbf{x})} \right)^2 & \text{if } \mathbf{y} \text{ and } \mathbf{x} \text{ are directly connected} \\ \left(\frac{\|\mathbf{y}-\mathbf{x}_c\|_2+\|\mathbf{x}_c-\mathbf{x}\|_2}{h(\mathbf{x})} \right)^2 & \text{if } \mathbf{y} \text{ and } \mathbf{x} \text{ are indirectly connected} \end{cases} \tag{4.32}$$

Remark: In the example above, since it is in 2D, we have

$$\mathbf{X}_c = \{ \ \mathbf{x}_c \ \} = \{ \ (0,0)^T \ \}$$

The next sketch shows the contour lines of the weighting function W^n around selected values for $\mathbf{y}$ in the neighborhood of the sharp edge of our example.



This setup for W^n provided some success when we were dealing with sharp edges of flat shaped bodies like paper or sheets. It guarantees that W^n keeps its smoothness properties, that means if $W^n(r)$ is smooth in r then also $W^n(r(\mathbf{y},\mathbf{x}))$ is smooth in $\mathbf{y}$.

Moreover, the generalized setup of W^n is not the only card we can bring into play. Another very important point is to drastically reduce the weights m_i in the neighborhood of $\mathbf{x}_c$. Without this, our computations showed that accurate results seemed to be impossible. We choose

$$m_i = 1 - e^{-\left(\frac{\|\mathbf{x}_i - \mathbf{x}_c\|_2}{\alpha\, h(\mathbf{x}_i)}\right)^2} \tag{4.33}$$

where $\alpha = \mathbf{O}(1)$ is a factor which can be varied a little bit to see, what value of α works best. Unfortunately, we were not able to give an exact value for α which works best.

In the appendix, the reader will find some computational results involving sharp edged geometries, which show that the method presented here works.

Chapter 5

5. Some Useful Technical Items

In this chapter we would like to have a look at some technical details that are related to the general SPH approach. We will speak about globally weighting particles by a weight m_i, which does not mean physical mass, but rather a weight, stating the 'importance' of some particle. This has some meaning when treating trailing edges (see section 5.1). Further on, we will speak about the problem of nearest neighbor search in the case of constant and also in the case of variable smoothing length (section 5.2). Often, we observed phenomena of particles being too far disjoint from each other or evenly particles being too close to each other compared to the smoothing length. This problem might be solved by either readapting the smoothing length to the particle density, or by adding/removing particles to/from the computation domain. This will be briefly considered in section 5.3.

5.1. Physical weight of particles

In section 3 we explained in detail how to compute a smoothed particles approximation $\Pi^n_{\omega_N} f$ of some function f for a given particle system $\omega_N(t)$. We saw that each particle i, being in the neighborhood of an arbitrary point $\mathbf{y} \in \Omega(t)$ obtains the weighting $W^n(\mathbf{y}, \mathbf{x}_i) \cdot m_i$. This weighting is a combination of local weighting ($W^n(\mathbf{y}, \mathbf{x}_i)$) depending on the distance between $\mathbf{y}$ and $\mathbf{x}_i$, and global weighting m_i providing 'physical' weight to particles. We did not consider the global weights yet. However, global weighting can be important, for example

- if shocks are involved in the computation. The particles upstream from the shock are sparsely distributed, whereas the particles downstream are much denser depending on how much compressing the shock is. In those cases it might be favorable to give a higher physical weight to the particles upstream than to the particles downstream in order to liberalize the particle density.
- if sharp edges are involved in the computation (see section 4.3). By experiment we found out that it is absolutely necessary to reduce the physical weight of the particles being close to some sharp edge, otherwise instabilities occurred

despite the tricky construction of W^n in the neighborhood of sharp edges.

The global weight of a particle i is $m_i(t)$ which is now a function of the time. Unlike classical SPH, m_i does not represent a piece of mass, but it represents the 'importance' of the particle for establishing the smoothed particle approximation Πf of some function f.

For occurring shocks, one could set $m_i := \frac{1}{\rho_i}$, even more precisely, $m_i := V_i$ where V_i is just the average volume which is taken by particle i, which obeys the differential equation

$$\frac{dV_i}{dt} = V_i \, \nabla \Pi^n_{\omega_N} \mathbf{v}_i$$

Also for sharp edges, it might be favorable to set the m_i. A few words were already given in section 4.3.

However, if no shocks or sharp edges occur, it is usually sufficient to set $m_i = 1 \ \forall i \in \{1, ..., N\}$.

5.2. Searching for Nearest Neighbors

The search for the surrounding particles of particle i at position $\mathbf{x}_i$ is one of the central problems which has to be solved very efficiently because this search has to be done for each particle during each time step. In this section, we will suggest methods for cases where the smoothing length h is constant throughout $\Omega(t)$ (section 5.2.1) as well as for cases where the smoothing length is a function in the sense $h = h(\mathbf{x})$ (section 5.2.2)

5.2.1. Nearest Neighbors for Constant Smoothing Length

Definition 5.1 (Surrounding Particles for constant h)
The set of particles surrounding the position $\mathbf{x}$ is given by

$$L_{n_p}(\mathbf{x}) := \{ \ i \mid \|\mathbf{x} - \mathbf{x}_i\|_2 \leq h \ ; \ i \in \{1, ..., N\} \ \} \tag{5.1}$$

$L_{n_p}(\mathbf{x})$ contains the indices of all particles for which we have $W(\mathbf{x}, \mathbf{x}_i) \neq 0 \quad \forall i \in L_{n_p}(\mathbf{x})$.

The technique is to sort all particles into a frame of cells with edge length h before executing a single time step (evaluation of the function $\mathbf{G}(\mathbf{Y})$); the cells are given successive numbers, such that we know what particles are in what cells. In order to find $L_{n_p}(\mathbf{x})$, we have to easily determine the number of the cell which the point $\mathbf{x}$ is placed in. The neighboring particles are among those particles which are in the cell of $\mathbf{x}$ and in all adjacent cells.

Before running a time step, we have to first sort the particles into the cells, which is described below.

- We establish a major box

  $$B := \left\{ \mathbf{x} \mid \mathbf{x}^- \leq \mathbf{x} < \mathbf{x}^+ \text{ componentwise} \right\}$$

  with the condition that $\mathbf{x}_i \in B \;\; \forall i \in \{1, ..., N\}$ and that

  $$\mathbf{x}^+ - \mathbf{x}^- = \mathbf{m} \cdot h$$

  with $\mathbf{m} \in \mathbf{N}^\nu$, $\mathbf{m} := (m^{(1)}, m^{(2)}, ...)^T$. $\mathbf{m}$ contains the number of single cells with edge length h, contained in B in each direction.

- We subdivide B into sub cells b_α, $\alpha \in \mathbf{N}^\nu$. b_α is of edge length h and is defined by

  $$b_\alpha := \left\{ \; \mathbf{x} \mid \mathbf{x}^- + \alpha \cdot h \leq \mathbf{x} < \mathbf{x}^- + (\alpha + \mathbf{1}) \cdot h \right\}$$

  Remark: It is very easy to determine the cell containing $\mathbf{x}$ by

  $$\alpha = int\left(\frac{\mathbf{x} - \mathbf{x}^-}{h} \right) \quad \text{componentwise}$$

- For the purpose of successive numbering of the cells we define

  $$b_{n_c} := b_\alpha \qquad \text{where} \quad n_c \in \mathbf{N} \;, \quad n_c \;=\; \alpha^{(1)} \;+\; \sum_{i=2}^{\nu} \left(\prod_{j=1}^{i-1} m^{(j)} \right) \cdot \alpha^{(i)}$$

  where n_c is the index of the cell. We define $n_c(i)$ to be the index of the cell containing particle i.

- We have to establish a list with particle indices and a corresponding list of cell indices:

  $$\begin{aligned} L_{n_p} &:= [1, 2, ...N] \\ L_{n_c} &:= [n_c(1), n_c(2), ...n_c(N)] \end{aligned}$$

- We have to reorder the two lists simultaneously such that the entries of L_{n_c} are ordered by size

$$\begin{aligned} L_{n_p} &:= [n_p(1), n_p(2), ...n_p(N)] \\ L_{n_c} &:= [n_c(n_p(1)), n_c(n_p(2)), ...n_c(n_p(N))] \end{aligned}$$

  where $n_p(i)$ denotes the value of the i-th entry in L_{n_p}. We have

$$n_c(n_p(i)) \leq n_c(n_p(i+1)) \qquad i = 1...N-1$$

  The reordering takes only $\mathbf{O}(N \cdot log_2(N)\,)$ operations (see [14]).

After the preparations above it has become very easy to compute a list of the indices of those particles which are placed in cell b_{n_c} for arbitrary n_c. Given n_c, we compute the indices of all particles in b_{n_c} by the procedure below.

- We search for the first appearance of the number n_c in the list L_{n_c} and call the position in the list $p_s(n_c)$.
  Remark: Since L_{n_c} is ordered by size, this search takes only $\mathbf{O}(log_2(N))$ operations (see [14]).

- We search for the last appearance of the number n_c in L_{n_c} and call that position $p_f(n_c)$.
  Remark: this search takes only as many operations as particles are contained in b_{n_c}.

- We establish the list of indices of all particles in b_{n_c} by

$$L_{n_p}(n_c) := \{\ n_p(i) \mid p_s \leq i \leq p_f\ \}$$

Now we are ready to look efficiently for neighboring particles to a point $\mathbf{x}$. The simple procedure is as follows.

- We establish α as defined above. This determines the cell that $\mathbf{x}$ is located in.

- We establish the list of all cell numbers surrounding $\mathbf{x}$.

$$L_{n_c}(\mathbf{x}) \ := \ \{\ n_c \mid n_c \ = \ (\alpha^{(1)} + \delta^{(1)}) \ + \ \sum_{i=2}^{\nu} \left(\prod_{j=1}^{i-1} m^{(j)} \right) \cdot (\alpha^{(i)} + \delta^{(i)})\ ; \\ \delta \in \{-1, 0, 1\}^{\nu}\ \}$$

- We establish the list of indices of all particles in the surrounding cells

$$\bar{L}_{n_p}(\mathbf{x}) := \bigcup_{n_c \in L_{n_c}(\mathbf{x})} L_{n_p}(n_c)$$

- $\bar{L}_{n_p}(\mathbf{x})$ is a conveniently small set for which we have

$$L_{n_p}(\mathbf{x}) \subset \bar{L}_{n_p}(\mathbf{x})$$

We see that finding the indices of particles surrounding a point $\mathbf{x}$ is actually not very expensive. We have described this work for cases of constant h. Now, lets have a look at the technique where h is not constant, but may vary in space.

5.2.2. Nearest Neighbors for Variable Smoothing Length

For variable smoothing length it is necessary to establish a function $h : \Omega(t) \to \mathbf{R}^+$. We require that

- $h(\mathbf{x})$ be differentiable
- $\|\nabla h\|_2 < 1 \quad \forall \mathbf{x} \in \Omega(t)$

We define $D_h := \max\limits_{\mathbf{x}\in\Omega(t)} (\|\nabla h(\mathbf{x})\|_2)$ and $h_0 := \min\limits_{\mathbf{x}\in\Omega(t)} (h(\mathbf{x}))$
Under these conditions finding nearest neighbors of a point $\mathbf{x} \in \Omega(t)$ is almost as easy as for finding neighbors for constant h.

Definition 5.2
We define the set of relevant neighbors of a point $\mathbf{x} \in \Omega(t)$ by

$$L_{n_p}(\mathbf{x}) := \{\ i \mid \|\mathbf{x} - \mathbf{x}_i\|_2 \leq h(\mathbf{x}_i)\ ;\ i \in \{1, ..., N\}\ \}$$

Under the two conditions on h assumed above we give the statement

$$h(\mathbf{x}_i) \in \left[h(\mathbf{x}) \cdot (1 - D_h)\ ,\ h(\mathbf{x}) \cdot \frac{1}{1 - D_h} \right] \qquad \forall i \in L_{n_p}(\mathbf{x})$$

This just means that a point $\mathbf{x}$ with smoothing length $h(\mathbf{x})$ has only neighbors whose smoothing length can be found in the interval $\left[h(\mathbf{x}) \cdot (1 - D_h)\ ,\ h(\mathbf{x}) \cdot \frac{1}{1 - D_h} \right]$, or it is itself neighbor to a particle whose smoothing length is in this interval.

This fact suggests to introduce discrete smoothing lengths for sorting particles into cells. The discrete smoothing lengths are

$$h_a := h_0 \cdot \frac{1}{(1 - D_h)^a} \qquad a = 0, 1, ...$$

Now it is easy to establish various cell systems with $h_a,\ a = 0, 1, ...$ being the associated cell edge length. Like for the case with constant smoothing length, the particles have to be sorted into the cells of the various cell systems. However, every single particle needs to be sorted into the appropriate cell of **only three** cell systems. The according smoothing lengths for particle j are h_{a-1}, h_a, h_{a+1} where a is chosen such that $h_{a-1} \leq h(\mathbf{x}_j) < h_a$. After this kind of sorting, all neighboring particles to particle j can easily be located. We turn to describing the necessary steps in detail. The steps are similar to those for the case of constant h, however here we will have to do it for a multi box system.

Before executing a single time step, we have to perform the sorting of all particles into the appropriate boxes.

- We establish major boxes

  $$B_a := \left\{ \mathbf{x} \mid \mathbf{x}_a^- \leq \mathbf{x} < \mathbf{x}_a^+ \text{ componentwise ; } \right\} \quad a = 0, 1, ...$$

  with the condition that $\mathbf{x}_i \in B_a \ \ \forall i = \{1, ..., N\}$, $\forall a \geq 0$. We have

  $$\mathbf{x}_a^+ - \mathbf{x}_a^- = \mathbf{m}_a \cdot h_a$$

  with $\mathbf{m}_a \in \mathbf{N}^\nu$, $\mathbf{m}_a := (m_a^{(1)}, m_a^{(2)}, ...)^T$. $\mathbf{m}_a$ contains the number of cells contained in B_a in each direction.

- We subdivide B_a into sub cells $b_{a,\alpha}$ of the edge length h_a, defined by

  $$b_{a,\alpha} := \left\{ \mathbf{x} \mid \mathbf{x}_a^- + \alpha \cdot h_a \leq \mathbf{x} < \mathbf{x}_a^- + (\alpha + \mathbf{1}) \cdot h_a \ ; \ \ a = 0, 1, ... \right\}$$

  Remark: It is very easy to determine the cell in the h_a-system containing $\mathbf{x}$ by

  $$\alpha = int\left(\frac{\mathbf{x} - \mathbf{x}_a^-}{h_a} \right) \qquad \text{componentwise}$$

- For the purpose of successive numbering of the cells we define

  $$b_{a,n_c} := b_{a,\alpha} \qquad \text{where} \quad n_c \ = \ \alpha^{(1)} \ + \ \sum_{i=2}^{\nu} \left(\prod_{j=1}^{i-1} m_a^{(j)} \right) \cdot \alpha^{(i)}$$

  We define $n_c(i)$ to be the number of the particular cell containing particle i.

- We have to establish ordered lists with particle numbers and a corresponding cell numbers:

  $$\begin{aligned} L_{a,n_p} &:= \ \{ \ i \mid h_{a-2} \leq h(\mathbf{x}_i) < h_{a+1} \ ; \ i \in \{1, ..., N\} \ \} \\ L_{a,n_c} &:= \ \{ \ n_c(i) \mid i \in L_{a,n_p} \ \} \end{aligned}$$

The first list is a list of indices of particles which are sorted into the cell system with edge length h_a. Remember, that each particle is sorted only into three different cell systems.

- We have to reorder the two lists simultaneously such that the entries of L_{a,n_c} are ordered by size

  $$\begin{aligned} L_{a,n_p} &:= [n_p(1), n_p(2), ..., n_p(N_a)] \\ L_{a,n_c} &:= [n_c(n_p(1)), n_c(n_p(2)), ..., n_c(n_p(N_a))] \end{aligned}$$

  where $n_p(i)$ denotes the value of the i-th entry in L_{a,n_p}, N_a is the number of elements in L_{a,n_p}. Since we ordered by size, we have

  $$n_c(n_p(i)) \leq n_c(n_p(i+1)) \qquad i = 1...N_a - 1$$

Now, like for constant smoothing length, we see it is very easy to find the indices of all particles which are placed in the cell b_{a,n_c} and whose smoothing length is in the interval $[\ h_{a-2}\ ,\ h_{a+1}\ )$. Given a and the index of the box, this is done in the following way.

- We search for the first appearance of the number n_c in the list L_{a,n_c} and call the position in the list $p_{a,s}(n_c)$.
  Remark: Since L_{a,n_c} is ordered by size, this search takes only $\mathbf{O}(log_2(N_a))$ operations

- We search for the last appearance of the number n_c in L_{a,n_c} and call that position $p_{a,f}(n_c)$.
  Remark: this search takes only as many operations as particles are contained in b_{a,n_c}.

- We establish the list of indices of all particles in b_{a,n_c} by

  $$L_{a,n_p}(n_c) := \{\ n_p(i) \mid p_{a,s}(n_c) \leq i \leq p_{a,f}(n_c)\ \}$$

After these preparations, to find the relevant neighboring particles to a point $\mathbf{x}$, we proceed as follows.

- We determine a such that $h_{a-1} \leq h(\mathbf{x}) < h_a$

- We establish α as defined above with respect to h_a. This determines the cell b_{a,n_c} which $\mathbf{x}$ is located in.

- We establish the list of indices of all cells surrounding $\mathbf{x}$.

  $$\begin{aligned} L_{a,n_c}(\mathbf{x}) \ := \ \{\ n_c \mid n_c \ = \ (\alpha^{(1)} + \delta^{(1)}) \ + \ \sum_{i=2}^{\nu} \left(\prod_{j=1}^{i-1} m_a^{(j)} \right) \cdot (\alpha^{(i)} + \delta^{(i)}) \ ; \\ \delta \in \{-1, 0, 1\}^{\nu}\ \} \end{aligned}$$

- We establish the list of indices of all particles in the surrounding cells

$$\bar{L}_{n_p}(\mathbf{x}) := \bigcup_{n_c \in L_{a,n_c}(\mathbf{x})} L_{a,n_p}(n_c)$$

- $\bar{L}_{n_p}(\mathbf{x})$ is a conveniently small set for which we have

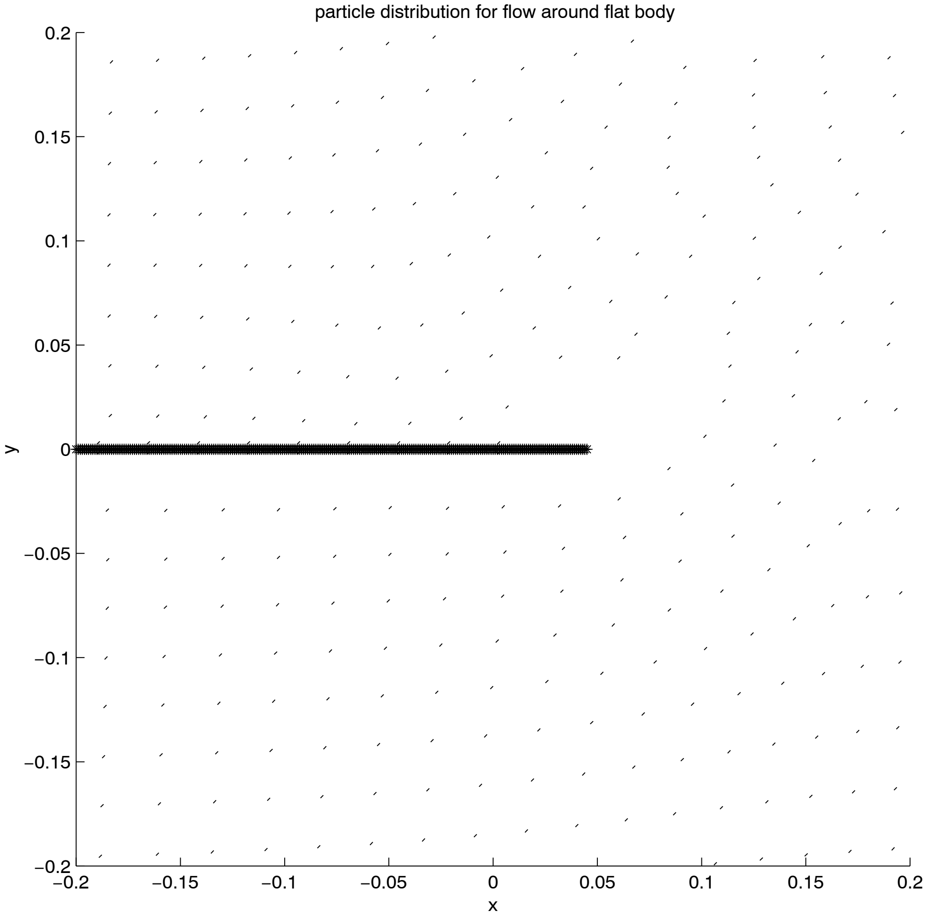
$$L_{n_p}(\mathbf{x}) \in \bar{L}_{n_p}(\mathbf{x})$$

Remark: We can be sure that, by the above introduced method, we capture all surrounding particles. Since neighbors of $\mathbf{x}$ have smoothing lengths in the interval $\left[h(\mathbf{x}) \cdot (1 - D_h)\ ,\ h(\mathbf{x}) \cdot \frac{1}{1-D_h}\right]$, we can be certain that all neighboring particles were at least sorted into the cell system with edge length h_a.

5.3. Adding and Removing Particles

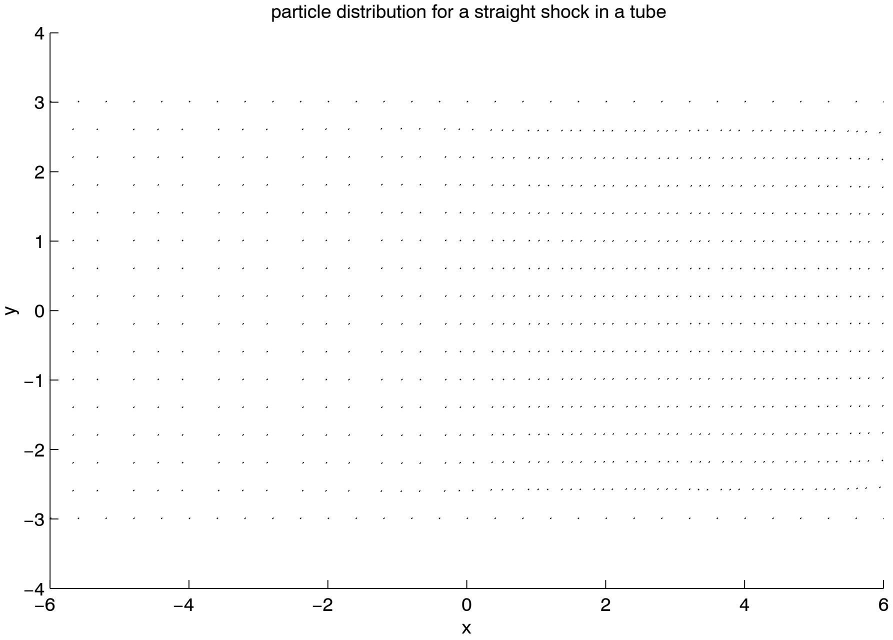
This section shows briefly the necessity of adding and removing particles. Since particles move with the fluid flow, they change the distance between each other steadily. Thereby it might happen, that large holes occur (area with a diameter of almost the smoothing length) or that particles are too dense. The next two pictures are to illustrate these effects.

1.) The following plot shows a snapshot of the particle distribution for a flow around the edge of a flat body (the flat body is the thick line at $y = 0$).



We see that in the neighborhood of the edge there are particles sparsely distributed. The occurring hole has a diameter of about the smoothing length which was $h = 0.06$ in this particular example. Such holes lead to a breakdown of the computation. One has the (traditional) chance to readapt the smoothing length in this area as suggested by Benz (see [8]). However, this would mean that the accuracy of approximation would drop because of rising smoothing length (refer to chapter 3, approximation properties). Moreover, it would mean that, eventually, one would have to handle a very complicated function $h(\mathbf{x})$. Another, more convenient way, that we would like to suggest here, is to introduce particles into regions where they are lacking.

2.) The following plot shows a snapshot of the particle distribution for a straight shock in a tube.



We easily recognize the concentration of particles for $x > 0$. The smoothing length was $h = 1$. Again, to save computation time, one could decrease the smoothing length in those regions where the particles are dense. This results in an (possibly not wanted) increase of computation accuracy and in the necessity of handling an eventually complicated function $h(\mathbf{x})$. Moreover, particles remain in a rather anisotropic distribution. The more convenient method, that we would like to suggest here, is to remove particles where they are too dense compared to the smoothing length.

5.3.1. Adding particles

Suppose, we add a new particle with index n at the position $\mathbf{x}_n$ at a certain time $t_{0,n}$. Then the particle needs to obtain a set of information Φ_n in order to take part in the computation. We simply chose

$$\Phi_n = \Pi^n_{\omega_N} \Phi(\mathbf{x}_n)$$

where the smoothed particle operator was chosen of high order, i.e. the least squares approximation polynomial $p_{\mathbf{y}}(\mathbf{x})$ was of degree $d \geq 2$. Actually, often it was necessary to smooth out the Φ_i of the yet existing particles in the neighborhood of $\mathbf{x}_n$ in the fashion

$$\Phi_i^{\mathrm{new}}(t_{0,n}) := \Phi_i(t_{0,n}) \cdot [\, 1 - W(r(\mathbf{x}_n, \mathbf{x}_i))\,] + \Pi^n_{\omega_N} \Phi(\mathbf{x}_i) \cdot W(r(\mathbf{x}_n, \mathbf{x}_i))$$

where W is the weight function, introduced in chapter 3. Also, the weight of the newly input particle $m_n(t)$ had to be increased by

$$m_n(t) = 1 - e^{-\frac{t - t_{0,n}}{a}} \quad ; \quad a = \frac{h}{c}$$

in order to smoothly involve the new particle. We found this to be necessary to avoid instabilities in the solution.

5.3.2. Removing Particles

Above, the advantages of removing particles were already discussed. Here, we briefly describe the technique of removing. Suppose, at some time $\bar{t}$, there are two particles with indices m and n whose distance between each other satisfies the condition

$$\| \, \mathbf{x}_n - \mathbf{x}_m \, \|_2 \; < \; a \cdot h$$

where a is a prescribed number (we chose $a := 0.2$). Then these particles are combined to one new particle with the index q, assuming the values

$$\mathbf{x}_q := \frac{\mathbf{x}_n + \mathbf{x}_m}{2}$$
$$\Phi_q := \frac{\Phi_n + \Phi_m}{2}$$

As for the case of adding particles, we found it to be necessary to smooth out the values of Φ of the particles in the neighborhood of $\mathbf{x}_q$ by

$$\Phi_i^{\mathrm{new}}(\bar{t}) := \Phi_i(\bar{t}) \cdot [\, 1 - W(r(\mathbf{x}_n, \mathbf{x}_i))\,] + \Pi^n_{\omega_N} \Phi(\mathbf{x}_i) \cdot W(r(\mathbf{x}_n, \mathbf{x}_i))$$

in order to avoid instabilities. The order of approximation was chosen as $d = 2$ for the $\Pi^n_{\omega_N}\Phi$. We did not found it necessary to treat the physical weight m_q in special.

Chapter 6

6. Upwinding the General Smoothed Particle Scheme

In this chapter, we would like to consider one technique to upwind the general smoothed particle scheme. We find this to be important for the treatment of shocks with SPH. In chapter 2, we found it necessary to have some numerical viscosity induced by the numerical time integration method, especially when treating shocks. The numerical viscosity, induced by the time integration method, is easily implemented by programming an ODE solver. However, this method has two important disadvantages. First, it is not really clear, how large the numerical viscosity should be in order to treat shocks properly. The numerical viscosity depends on the size of the time step. Second, the time integration cannot be but of first order (concerning the time step size). This for example might lead to the phenomenon, that, when computing Navier-Stokes, the numerical viscosity is much larger than the viscous effects of the fluid. So, we would like to be able to perform the time integration in higher order, but we still like to be able to treat shocks.

Therefore, we introduce a method to upwind the general smoothed particle scheme. As we will see, this also induces a certain kind of numerical viscosity, but it has two important advantages. First, the numerical viscosity will not depend on the size of the time step. Second, we will be able to give an approximation of the numerical viscosity in terms of physical viscosity.

The steps of introducing the upwinding is to derive an upwind smoothed particle scheme from the Euler equations in characteristic form (done in section 6.1). The results of section 6.1 are then incorporated in the general smoothed particle scheme as given in chapter 2 (see section 6.2). In section 6.3 and 6.4 we are going to involve certain simplifications such that the method is easily applicable for computer programs with minimal effort.

6.1. Upwind Particle Scheme Based on Characteristic Theory

In this section we will introduce a particle scheme based on the characteristic formulation of the Euler equations. We would like to refer to chapter 4, where we showed in detail how to derive the characteristic formulation of the Euler equations for compressible fluid flow. First, let us outline the result of all the computations of section 4.1 again at this point, see equation (4.28):

$$
\begin{aligned}
c^2\frac{d\rho}{dt} + \rho c n^{(3)}\frac{du^{(2)}}{dt} - \rho c n^{(2)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(2)}\frac{\partial p}{\partial x^{(3)}} - c n^{(3)}\frac{\partial p}{\partial x^{(2)}} \\
c^2\frac{d\rho}{dt} - \rho c n^{(3)}\frac{dv^{(1)}}{dt} + \rho c n^{(1)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(3)}\frac{\partial p}{\partial x^{(1)}} - c n^{(1)}\frac{\partial p}{\partial x^{(3)}} \\
c^2\frac{d\rho}{dt} + \rho c n^{(2)}\frac{dv^{(1)}}{dt} - \rho c n^{(1)}\frac{dv^{(2)}}{dt} - \frac{dp}{dt} &= c n^{(1)}\frac{\partial p}{\partial x^{(2)}} - c n^{(2)}\frac{\partial p}{\partial x^{(1)}} \\
\rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= \rho c^2\,\nabla\mathbf{v} - c(\nabla p\cdot\mathbf{n}) \\
\rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} + \frac{dp}{dt} &= -\rho c^2\,\nabla\mathbf{v} - c(\nabla p\cdot\mathbf{n})
\end{aligned}
\tag{6.1}
$$

In equations (6.1), for simplicity, we omitted the source terms $\mathbf{l}_i \cdot S$. Equations (6.1) form the system of Euler equations in characteristic form, each of which representing a piece of characteristic information traveling at some characteristic speed along a characteristic surface. From chapter 4 we remember that

$$\mathbf{n} := (\ n^{(1)}, n^{(2)}, n^{(3)}\ )^T \quad \text{with} \quad ||\mathbf{n}||_2 = 1$$

represents a direction into which we would like to trace the flow of information. The five equations above are called compatibility relations. The characteristic speed for the first three compatibility relations is

$$\lambda_1 = \lambda_2 = \lambda_3 = \mathbf{v}^T \cdot \mathbf{n} := v_n$$

The characteristic speeds of the fourth and fifth compatibility relations is

$$
\begin{aligned}
\lambda_4 &= \mathbf{v}^T \cdot \mathbf{n} - c = v_n - c \\
\lambda_5 &= \mathbf{v}^T \cdot \mathbf{n} + c = v_n + c
\end{aligned}
$$

Now we would like to set up a numerical scheme based on the compatibility relations. Consider a particle at the position $\mathbf{y}$. We know that from point $\mathbf{y}$, three pieces of characteristic information travel with the speed v_n into the direction $\mathbf{n}$. After a representative amount of time $\Delta\tau$, the characteristic information arrived at the position

$$\bar{\mathbf{y}} = \mathbf{y} + \Delta\tau\ v_n\mathbf{n} \tag{6.2}$$

At this point $\bar{\mathbf{y}}$, two other pieces of characteristic information arrive, however not from the point $\mathbf{y}$ but from different positions. The fourth characteristic information comes in from point $\mathbf{y}^-$ for which we have

$$\begin{aligned} \bar{\mathbf{y}} &= \mathbf{y}^- + \Delta\tau\ \lambda^{(4)}\mathbf{n} \quad \leftrightarrow \\ \mathbf{y} + v_n\Delta\tau &= \mathbf{y}^- + \Delta\tau\ (v_n^- - c^-)\mathbf{n} \quad \leftrightarrow \\ \mathbf{y}^- &= \mathbf{y} + \Delta\tau\,(v_n - v_n^-)\mathbf{n} + \Delta\tau\, c^-\mathbf{n} \end{aligned} \tag{6.3}$$

where
$c^- := c(\mathbf{y}^-)$ is the sonic speed at position $\mathbf{y}^-$, and
$v_n^- := \mathbf{v}^T(\mathbf{y}^-)$ is the velocity in $\mathbf{n}$-direction at position $\mathbf{y}^-$

The fifth piece of characteristic information comes in from point $\mathbf{y}^+$ for which we have

$$\begin{aligned} \bar{\mathbf{y}} &= \mathbf{y}^+ + \Delta\tau\ \lambda^{(5)}\mathbf{n} \quad \leftrightarrow \\ \mathbf{y} + v_n\Delta\tau &= \mathbf{y}^+ + \Delta\tau\ (v_n^+ + c^+)\mathbf{n} \quad \leftrightarrow \\ \mathbf{y}^+ &= \mathbf{y} + \Delta\tau\,(v_n - v_n^+)\mathbf{n} - \Delta\tau\, c^+\mathbf{n} \end{aligned} \tag{6.4}$$

where
$c^+ := c(\mathbf{y}^+)$ is the sonic speed at position $\mathbf{y}^+$, and
$v_n^+ := \mathbf{v}^T(\mathbf{y}^+)$ is the velocity in $\mathbf{n}$-direction at position $\mathbf{y}^+$

Now the setting of some particle scheme based on the characteristics is very easy. We would like to approximate the first three compatibility relations at the position $\mathbf{y}$ and, consequently, we have to approximate the fourth and fifth compatibility relations at the positions $\mathbf{y}^-$ and $\mathbf{y}^+$ respectively. Hence, the particle approximation of equations (6.1) looks like

$$\begin{aligned} c^2\frac{d\rho}{dt} + \rho c n^{(3)}\frac{du^{(2)}}{dt} - \rho c n^{(2)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(2)}\frac{\partial \Pi p}{\partial x^{(3)}} - c n^{(3)}\frac{\partial \Pi p}{\partial x^{(2)}} \\ c^2\frac{d\rho}{dt} - \rho c n^{(3)}\frac{dv^{(1)}}{dt} + \rho c n^{(1)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= c n^{(3)}\frac{\partial \Pi p}{\partial x^{(1)}} - c n^{(1)}\frac{\partial \Pi p}{\partial x^{(3)}} \\ c^2\frac{d\rho}{dt} + \rho c n^{(2)}\frac{dv^{(1)}}{dt} - \rho c n^{(1)}\frac{dv^{(2)}}{dt} - \frac{dp}{dt} &= c n^{(1)}\frac{\partial \Pi p}{\partial x^{(2)}} - c n^{(2)}\frac{\partial \Pi p}{\partial x^{(1)}} \\ \rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} - \frac{dp}{dt} &= \rho c^2\, \nabla\Pi\mathbf{v}^- - c(\nabla\Pi p^- \cdot \mathbf{n}) \\ \rho c n^{(1)}\frac{dv^{(1)}}{dt} + \rho c n^{(2)}\frac{dv^{(2)}}{dt} + \rho c n^{(3)}\frac{dv^{(3)}}{dt} + \frac{dp}{dt} &= -\rho c^2\, \nabla\Pi\mathbf{v}^+ - c(\nabla\Pi p^+ \cdot \mathbf{n}) \end{aligned} \tag{6.5}$$

This represents a linear system of equations with the total time derivatives of the primitive variables ρ, $\mathbf{v}$ and p being the unknowns. The solution of this system is given by

$$
\begin{aligned}
\frac{d\rho}{dt} &= -\rho\,\frac{1}{2}\left(\nabla\Pi\mathbf{v}^{+} + \nabla\Pi\mathbf{v}^{-}\right) \;-\; \frac{1}{2c}\left[\,\nabla\Pi\left(\,p^{+} - p^{-}\,\right)\,\right]\cdot\mathbf{n} && (6.6)\\
\frac{d\mathbf{v}}{dt} &= -\frac{1}{\rho}\,\nabla\Pi p \;-\; \frac{1}{2\rho}\left[\,\mathbf{n}\cdot\nabla\Pi\left(p^{+} - 2p + p^{-}\right)\,\right]\cdot\mathbf{n} \;-\\
&\quad \frac{c}{2}\;\left[\,\nabla\Pi\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\,\right]\cdot\mathbf{n} && (6.7)\\
\frac{dp}{dt} &= -\rho c^{2}\,\frac{1}{2}\left(\nabla\Pi\mathbf{v}^{+} + \nabla\Pi\mathbf{v}^{-}\right) \;-\; \frac{c}{2}\left[\,\nabla\Pi(p^{+} - p^{-})\,\right]\cdot\mathbf{n} && (6.8)
\end{aligned}
$$

This is a numerical scheme. We would like to simplify this scheme by neglecting some terms, that are small. In this way, we obtain the following set of simplified equations

$$
\begin{aligned}
\frac{d\rho}{dt} &\approx -\rho\,\nabla\Pi\mathbf{v} \;-\; \frac{1}{2c}\left[\,\nabla\Pi\left(p^{+} - p^{-}\right)\,\right]\cdot\mathbf{n} && (6.9)\\
\frac{d\mathbf{v}}{dt} &\approx -\frac{1}{\rho}\,\nabla\Pi p \;-\; \frac{c}{2}\;\left[\,\nabla\Pi\left(\mathbf{v}^{+} - \mathbf{v}^{-}\right)\,\right]\cdot\mathbf{n} && (6.10)\\
\frac{dp}{dt} &\approx -\rho c^{2}\,\nabla\Pi\mathbf{v} \;-\; \frac{c}{2}\left[\,\nabla\Pi(p^{+} - p^{-})\,\right]\cdot\mathbf{n} && (6.11)
\end{aligned}
$$

Another simplification is necessary concerning the positions $\mathbf{y}^{+}$ and $\mathbf{y}^{-}$. Since the laws of computing these two positions (6.3) and (6.4) are implicit laws, for convenience we simplify

$$
\begin{aligned}
\mathbf{y}^{-} &= \mathbf{y} + \Delta\tau\, c\mathbf{n} && (6.12)\\
\mathbf{y}^{+} &= \mathbf{y} - \Delta\tau\, c\mathbf{n} && (6.13)
\end{aligned}
$$

which assumes, that the differences of the fluid velocities and the sound speeds between the different positions $\mathbf{y}$, $\mathbf{y}^{+}$, and $\mathbf{y}^{-}$ are negligible.

It is still not discussed yet how to choose the direction vector $\mathbf{n}$ and the representative amount of time $\Delta\tau$.

For $\mathbf{n}$, we choose the direction where changes of the flow field most probably come from:

$$
\mathbf{n} := \begin{cases} \frac{\nabla\Pi p}{\|\nabla\Pi p\|_2} & \text{if } \|\nabla\Pi p\|_2 \neq 0 \\ \mathbf{e}_1 & \text{otherwise} \end{cases} \qquad (6.14)
$$

Remark: For the case of $\|\nabla\Pi p\|_2 = 0$, no direction of fluid flow is dominant, hence for this case, not only $\mathbf{e}_1$, but any vector can be assigned to $\mathbf{n}$.

For $\Delta\tau$ we will give a formulation in section 6.3. At this point, we only would like to make clear that the order of magnitude of $\Delta\tau$ must be the same as the one of the term $\frac{h}{c}$ (which is the amount of time the sound needs to travel a distance of h).

Obviously, with equations (6.9)-(6.13), the construction of a numerical particle scheme based on the characteristic formulation of the Euler equations is already concluded. This scheme is in primitive variable form. As a disadvantage, the scheme requires that the values of $\nabla\Pi\mathbf{v}$ and $\nabla\Pi p$ be computed at the three different points $\mathbf{y}$, $\mathbf{y}^+$, and $\mathbf{y}^-$. This procedure seems to be computationally much more expensive than the general particle scheme introduced in chapter 2 because there we needed evaluations of $\nabla\Pi\mathbf{v}$ and $\nabla\Pi p$ only at one point. Moreover, this scheme is based on the primitive variable formulation, but the numerical method, introduced in chapter 2 is based on the conservative variable formulation.

Hence, in the next section we will transform the scheme (6.9)-(6.13) into a formulation in conservation variable form. We shall see, that the so derived numerical scheme fits easily into the general smoothed particle formulation given in chapter 2. In the sections thereafter we shall see, how to decrease the computational cost for this scheme considerably.

6.2. Upwind Smoothed Particle Scheme in Conservative Variable Formulation

Under the assumption that the changes of c are small compared to the changes of the pressure and velocities, we can rewrite (6.9)-(6.11) as

$$\frac{d\rho}{dt} = -\rho\left[\nabla\Pi\mathbf{v} + \nabla\Pi\left(\frac{1}{2\rho c}\left(p^+ - p^-\right)\right)\cdot\mathbf{n}\right] \tag{6.15}$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\left[\nabla\Pi p + \nabla\Pi\left(\frac{\rho c}{2}\left(\mathbf{v}^+ - \mathbf{v}^-\right)\right)\cdot\mathbf{n}\right] \tag{6.16}$$

$$\frac{dp}{dt} = -\rho c^2\left[\nabla\Pi\mathbf{v} + \nabla\Pi\left(\frac{1}{2\rho c}\left(p^+ - p^-\right)\right)\cdot\mathbf{n}\right] \tag{6.17}$$

which, in another writing, leads to

$$\begin{aligned}
\frac{d\rho}{dt} &= -\rho\, \nabla\Pi \mathbf{v}_{uw} \\
\frac{d\mathbf{v}}{dt} &= -\frac{1}{\rho}\, \nabla\Pi\, (p\, \mathbf{I} - \mathbf{St}_{uw}) \\
\frac{dp}{dt} &= -\rho c^2\, \nabla\Pi \mathbf{v}_{uw}
\end{aligned}$$

where the new involved velocity is

$$\mathbf{v}_{uw} := \mathbf{v} + \frac{1}{2\rho c}\, (p^+ - p^-)\mathbf{n} \tag{6.18}$$

and the new matrix is

$$\mathbf{St}_{uw}^{(i,j)} = -\frac{\rho c}{2} \cdot \left(v^{(i)^+} - v^{(i)^-} \right) n^{(j)} \tag{6.19}$$

Note that the index 'uw' stands for 'upwind'. The change from the primitive formulation (6.15)-(6.17) to the conservative variable formulation brings

$$\begin{aligned}
\frac{d\rho}{dt} &= -\rho\, \nabla\Pi \mathbf{v}_{uw} \\
\frac{d(\rho\mathbf{v})}{dt} &= -(\rho\mathbf{v})\, \nabla\Pi \mathbf{v}_{uw} \;-\; \nabla\Pi\, (p \;- \mathbf{St}_{uw}) \\
\frac{d(\rho E)}{dt} &= -(\rho E)\, \nabla\Pi \mathbf{v}_{uw} \;-\; p\, \nabla\Pi \mathbf{v}_{uw} \;-\; \nabla\Pi\, (p\, \mathbf{I} - \mathbf{St}_{uw}) \cdot \mathbf{v}
\end{aligned}$$

We recognize that the equations for ρ and $\rho\mathbf{v}$ are in conservation form, but not the one for ρE. However, in this case it is natural and also makes sense from a physical point of view to write down the upwind scheme for the total energy as

$$\frac{d(\rho E)}{dt} = -(\rho E)\, \nabla\Pi \mathbf{v}_{uw} \;-\; \nabla\Pi\, [\, (p\, \mathbf{I} - \mathbf{St}_{uw}) \cdot \mathbf{v}_{uw}]$$

which is now in conservation form. With this, the upwind particle scheme in conservation variables reads as

$$\frac{d\Phi}{dt} = -\Phi\, \nabla\Pi \mathbf{v}_{uw} - \nabla\Pi \mathbf{F}_{uw}^{Eu} + S \tag{6.20}$$

$$\frac{d\mathbf{x}}{dt} = \Pi \mathbf{v}_{uw} \tag{6.21}$$

where

$$\mathbf{F}_{uw}^{Eu} = (\; \mathbf{0}\, ,\, p\, \mathbf{I} - \mathbf{St}_{uw}\, ,\, (p\mathbf{I} - \mathbf{St}_{uw}) \cdot \mathbf{v}_{uw}\;) \tag{6.22}$$

is the usual flux function $\mathbf{F}^{Eu}$ extended by the presence of a stress tensor induced through upwinding.

Equations (6.20)-(6.21) represent the upwind particle scheme in conservative variables. It fits easily in the general smoothed particle scheme as introduced in chapter 2, equations (2.10/2.11). For the upwind method we apply the smoothed particle operator to an upwind velocity $\mathbf{v}_{uw}$ instead of the fluid velocity $\mathbf{v}$, and to an upwind flux function $\mathbf{F}^{Eu}_{uw}$ instead of the Euler flux function $\mathbf{F}^{Eu}$. Hence, all considerations made in chapter 2 about conservation properties and error estimation are indeed true even for the upwind scheme.

For Navier-Stokes flow, we claim that we can write down the flux function as

$$\mathbf{F}^{NS}_{uw} = \mathbf{F}^{Eu}_{uw} + (\mathbf{0}\ ,\ -\eta\,\mathbf{St}\ ,\ \mathbf{St}\cdot\mathbf{v}_{uw})$$

The disadvantage of this scheme is, again, that we would have to compute smoothed particle approximations of the velocity and of the flux function at the three different positions $\mathbf{y}$, $\mathbf{y}^+$, and $\mathbf{y}^-$. This costs three times as much computation time as the scheme presented in chapter 2 would require. In the next section we are going to propose a way how to decrease computation time.

6.3. Fast Upwind General Smoothed Particle Scheme

As we saw in the last section, the upwind particle scheme introduced is computationally very expensive because we have to evaluate the smoothed particle approximations at three different positions. In the present section we would like to look for a possibility to have a upwind particle scheme that requires function evaluations at only one representative position.

The possibility we suggest is to give first order approximations of the values $\left(v^{(i)^+} - v^{(i)^-}\right)$, $i = 1, ..., \nu$ and $(p^+ - p^-)$, based on a Taylor series expansion. This gives

$$\begin{aligned} \left(v^{(i)^+} - v^{(i)^-}\right) &= -2\,(\nabla v^{(i)}\cdot\mathbf{n})\,c\,\Delta\tau \;+\; \mathbf{O}(c^2\,\Delta\tau^2)\ , \quad i = 1...\nu \qquad (6.23)\\ \left(p^+ - p^-\right) &= -2\,(\nabla p\cdot\mathbf{n})\,c\,\Delta\tau \;+\; \mathbf{O}(c^2\,\Delta\tau^2) \qquad (6.24)\end{aligned}$$

and since we are not able to compute the exact derivatives $\nabla v^{(i)}\ ,\ \ i = 1...\nu$ and ∇p, we employ the approximative derivatives $\nabla\Pi v^{(i)}\ ,\ \ i = 1...\nu$ and $\nabla\Pi p$, and so, we obtain

$$\begin{aligned}\left(v^{(i)^+}-v^{(i)^-}\right) &\approx -2\left(\nabla\Pi v^{(i)}\cdot\mathbf{n}\right)c\,\Delta\tau \quad , \qquad i=1...\nu &(6.25)\\ \left(p^+-p^-\right) &\approx -2\left(\nabla\Pi p\cdot\mathbf{n}\right)c\,\Delta\tau &(6.26)\end{aligned}$$

We introduce these approximations into the expressions for $\mathbf{v}_{uw}$ and $\mathbf{St}_{uw}$. Thus we obtain

$$\begin{aligned}\bar{\mathbf{v}}_{uw} &:= \mathbf{v}-\frac{1}{\rho}\left[\,\nabla\Pi p\cdot\mathbf{n})\,\right]\cdot\mathbf{n}\,\Delta\tau\\ &= \mathbf{v}-\frac{\Delta\tau}{\rho}\nabla\Pi p &(6.27)\end{aligned}$$

$$\bar{\mathbf{St}}_{uw}^{(i,j)} := \rho c^2\Delta\tau\;\cdot\;\left(\nabla\Pi v^{(i)}\cdot\mathbf{n}\right)n^{(j)} \tag{6.28}$$

Finally, the fast upwind particle scheme appears as

$$\begin{aligned}\frac{d\Phi}{dt} &= -\Phi\,\nabla\Pi\bar{\mathbf{v}}_{uw}-\nabla\Pi\bar{\mathbf{F}}_{uw}+S &(6.29)\\ \frac{d\mathbf{x}}{dt} &= \Pi\bar{\mathbf{v}}_{uw} &(6.30)\end{aligned}$$

where

$$\bar{\mathbf{F}}_{uw}^{Eu} := \left(\,\mathbf{0}\,,\,p\,\mathbf{I}-\bar{\mathbf{St}}_{uw}\,,\,(p\,\mathbf{I}-\bar{\mathbf{St}}_{uw})\cdot\bar{\mathbf{v}}_{uw}\,\right)$$

and

$$\bar{\mathbf{F}}_{uw}^{NS} = \bar{\mathbf{F}}_{uw}^{Eu}+(\mathbf{0}\;,\;-\eta\,\mathbf{St}\;,\;\mathbf{St}\cdot\bar{\mathbf{v}}_{uw})$$

We see that with this scheme we do not have to compute the particle approximations of velocity and pressure at three different positions, but it suffices to do it at the position $\mathbf{y}$ where the particle under consideration is situated.

Finally, we want to say some words about choosing the representative amount of time $\Delta\tau$. We did not come up with a convenient theoretical solutions to this. Thus, we tried practically to give an expression

$$c\,\Delta\tau = k_1\left(\frac{h\,\nabla\Pi\bar{\mathbf{v}}_{uw}}{c}\right)^{k_2}\cdot h \tag{6.31}$$

To give values for k_1 and k_2 we tried to solve shocks of different strengths with the general upwind particle scheme and therewith, we tried to adjust the constants

k_1 and k_2 such that the numerical viscosity provided by the upwind terms is large enough to stabilize the shocks. The values we found for air are

$$k_1 = 0.3909 \tag{6.32}$$

$$k_2 = 0.4515 \tag{6.33}$$

In our appendix, we are going to exhibit some results obtained when computing shocks. The results show the upwind scheme is able to work well.

6.4. An Even Faster Upwind Particle Scheme

Section 6.3. introduces a fast upwinding scheme for the general smoothed particle ansatz. However, it is still somehow expensive. This is why in addition to the usual evaluations for Euler equations without upwinding (in that case evaluations have to be done for $\nabla\Pi\mathbf{v}_{uw}$, $\nabla\Pi p$, and $\nabla\Pi(p\,\mathbf{v}_{uw})$) now other evaluations have to be done, namely the ones connected with the matrix $\bar{\mathbf{S}}\mathbf{t}_{uw}$, equation (6.28). The additional evaluations are $\nabla\Pi\left(\nabla\Pi v^{(i)}\right),\ i = 1, ..., \nu$. Since the evaluation of the smoothed particle approximation is the computationally most expensive part, we are always interested in reducing the number of evaluations of the smooth particle approximation. Furthermore, we would expect the matrix $\bar{\mathbf{S}}\mathbf{t}_{uw}$ to be symmetric. However, in (6.28) it is not symmetric. In this section now we would like to show, how to derive some symmetric matrix of the same type.

As we saw, matrix $\bar{\mathbf{S}}\mathbf{t}_{uw}$ in equation (6.28) was derived from matrix $\mathbf{St}_{uw}$ in equation (6.19) by Taylor series expansion of the entries. However, it is also possible to simplify directly equation (6.16). With

$$\nabla\Pi\mathbf{v}^+ - \nabla\Pi\mathbf{v}^- \;=\; -2\,[\,\nabla\Pi\,(\,\nabla\Pi\mathbf{v}\,)\,]\cdot\mathbf{n}\,c\,\Delta\tau \tag{6.34}$$

we obtain directly

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\,\nabla\Pi p \;+\; \Big[\,\Big[\,\nabla\Pi\,\Big(\,c^2\,\nabla\Pi\mathbf{v}\,\Delta\tau\,\Big)\,\Big]\cdot\mathbf{n}\,\Big]\cdot\mathbf{n} \tag{6.35}$$

Equation (6.35) written in matrix vector expansion gives

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\,\nabla\Pi\left(\,p\,\mathbf{I} - \tilde{\mathbf{S}}\mathbf{t}_{uw}\,\right) \tag{6.36}$$

where

$$\tilde{\mathbf{S}}\mathbf{t}_{uw}^{(i,j)} = \rho c^2\,\nabla\Pi\mathbf{v}\,\Delta\tau\;n^{(i)}n^{(j)}$$

Now it is obvious that the matrix $\tilde{\mathbf{St}}_{uw}$ is symmetric as one would expect for Navier-Stokes like viscosity terms. We are able to readily write down the upwind numerical scheme, derived in this way.

$$\begin{aligned} \frac{d\Phi}{dt} &= -\Phi\,\nabla\Pi\tilde{\mathbf{v}}_{uw} - \nabla\Pi\tilde{\mathbf{F}}_{uw} + S && (6.37) \\ \frac{d\mathbf{x}}{dt} &= \Pi\tilde{\mathbf{v}}_{uw} && (6.38) \end{aligned}$$

where

$$\tilde{\mathbf{v}}_{uw} = \bar{\mathbf{v}}_{uw}$$

$$\tilde{\mathbf{F}}^{Eu}_{uw} := \Big(\,\mathbf{0}\,,\,p\,\mathbf{I} - \tilde{\mathbf{St}}_{uw}\,,\,(p\,\mathbf{I} - \tilde{\mathbf{St}}_{uw})\cdot\tilde{\mathbf{v}}_{uw}\,\Big)$$

and

$$\tilde{\mathbf{F}}^{NS}_{uw} = \tilde{\mathbf{F}}^{Eu}_{uw} + (\mathbf{0}\;,\;-\eta\,\mathbf{St}\;,\;\mathbf{St}\cdot\tilde{\mathbf{v}}_{uw})$$

For the scheme presented in the present section, additional evaluations are necessary only for $\nabla\Pi\,(\nabla\Pi\mathbf{v})$, compared to the basic scheme (2.10/2.11). So, in comparison to section 6.3., the scheme presented here runs even faster on the computer since the number of evaluations of the smoothed particle approximation is less.

For the value

$$c\,\Delta\tau = k_1\,\left(\frac{h\,\nabla\Pi\bar{\mathbf{v}}_{uw}}{c}\right)^{k_2}\cdot h \tag{6.39}$$

k_1 and k_2 are the same as in section 6.3.

Conclusion

With this thesis we exhibit a general SPH scheme, which was basically developed to overcome some disadvantages of the classical SPH. A big success was done for the treatment of boundaries of any kind by the involvement of the characteristic form of the Euler equations as well as by using a least squares particle approximation technique which provides good approximation properties at the boundary. Moreover, with the employment of a least squares approach it is possible to provide very good approximation properties (of different orders) in general. The general SPH approach is able to work easily with a smoothing length varying in space, which is also quite favorable for many applications. With the possibility of adding and removing particles, the method works quite stable since the appearance of 'particle holes' as well as of 'particle lumps' are avoided. Finally, with some upwinding technique, general SPH is also able to accurately compute shocks.

However, this thesis is only a single step in the direction of accurate and efficient particle computations for Euler and Navier-Stokes equations. A lot of interesting questions remain open. Some of them were already asked in this thesis. One very important questions is concerning the time integration of the general SPH scheme for low Mach number flow (see section 2.4.3.). This question is important because many complicated industrial problems (for example fluttering paper sheets in printing machines, wind mills) are based on a very low Mach number, but propose to be treated with an SPH scheme because of time dependent geometry.
Another question is, how to speed up the method. Especially the computation of the smoothed particle approximation of some function requires a lot of computation time. This is due to the large number of neighboring particles to be used. For two dimensional applications, the average number of neighboring particles is about 25, in three dimensions it is nearly a hundred.
The main task for the future as to involve SPH for computations of multi phase or free surface flows. General SPH should have some considerable advantages for those cases since it is a Lagrangian method.

Used Symbols and Abbreviations

Symbol	**Meaning**
$\mathbf{1}$	vector in $\mathbf{R}^\nu$, $\mathbf{1} := (1, 1, ...)^T$
c	speed of sound; section 4.1
$C^n(\Omega(t))$	space of n-times continuously differentiable functions defined on $\Omega(t)$
$\mathbf{e}_i$	unit vectors, $i \in \{1, ..., \nu\}$
E	total energy; $dE := du + \mathbf{v}^T \cdot d\mathbf{v}$, definition 2.4
$E\Pi f$	approximation error of function f; definition 2.2
$\mathbf{F}$	flux function; definition 2.3
$\mathbf{F}^{EU}$	flux function for Euler flow, definition 2.4
$\mathbf{F}^{EU}_{uw}$	upwind flux function for Euler flow, section 6.2
$\mathbf{F}^{NS}$	flux function for Navier-Stokes flow, definition 2.4
$\mathbf{g}$	gravity vector; $\mathbf{g} := (\ g^{(1)}, g^{(2)}, g^{(3)})^T\ )$; $g^{(i)}$ is gravity component in $\mathbf{e}^{(i)}$-direction; definition 2.4
$\mathbf{g}_i$	function providing the total time derivative of $\mathbf{y}_i$; eqn. (2.17)
$\mathbf{G}$	collection of time derivatives of particle information; $\mathbf{G} := \{\ \mathbf{g}_i \mid i \in \{1; ...; N\}\ \}$, section 2.2.
$h(\mathbf{x})$	smoothing length depending on $\mathbf{x} \in \Omega(t)$
h_i	smoothing length at position $\mathbf{x}_i$
$\mathbf{I}$	identity matrix
$\mathbf{l}^{(i)}$	i-th left eigenvector; section 4.1
m_i	'weight' of the particle with index i
$\mathbf{n}$	inward pointing normal to the boundary; $\mathbf{n} := (\ n^{(1)}, n^{(2)}, n^{(3)})^T\ )$
$\mathbf{n}$	direction into which information is traced; $\lVert\mathbf{n}\rVert_2 = 1$
N	number of particles in the flow domain $\Omega(t)$
p	pressure acting in the fluid
$p_\mathbf{y}(\mathbf{x})$	local least squares approximation at position $\mathbf{y} \in \Omega(t)$
$p(\mathbf{y}, \mathbf{x})$	local least squares approximation at position $\mathbf{y} \in \Omega(t)$
$r(\mathbf{y}, \mathbf{x})$	distance function between two geometrical locations $\mathbf{y}$ and $\mathbf{x}$, def. 3.1
$\mathbf{R}$	set of all real numbers
$\mathbf{R}^+$	set of all positive real numbers including zero
$\mathbf{R}^\nu$	set of all real vectors in ν dimensions
$S(\mathbf{x})$	source function; definition 2.3
S_i	source term at position $\mathbf{x}_i$
$\mathbf{St}$	deformation tensor; definition 2.4
$\mathbf{St}_{uw}$	upwind deformation tensor; section 6.2
T	time space; $T := \mathbf{R}^+$
T	total temperature of the fluid
$\mathbf{T}_{\Omega_0}(t)$	subdomain of fluid flow; definition 2.3

t	time
u	internal energy given by $du = c_v\, dT + \frac{p}{\rho^2}\, d\rho$
$\mathbf{v}$	velocity vector $\mathbf{v} := (v^{(1)}, v^{(2)}, v^{(3)})^T$ $v^{(i)}$ is the velocity component in $\mathbf{e}^{(i)}$-direction
$\mathbf{V}$	ensemble of primitive variables, section 4.1
$\mathbf{v}_{uw}$	upwind velocity; section 6.2;
W_h	smoothing kernel for classical SPH
$W^n(r)$	weighting function for general SPH; W^n is n-times continuously differentiable in r; definition 3.1
$\mathbf{x}$	geometrical position $\mathbf{x} := (x^{(1)}, x^{(2)}, x^{(3)})^T$ $x^{(i)}$ is the component in $\mathbf{e}^{(i)}$-direction
$\mathbf{x}_i$	position of the particle with index i
$\mathbf{y}$	geometrical position $\mathbf{y} := (y^{(1)}, y^{(2)}, y^{(3)})^T$ $y^{(i)}$ is the component in $\mathbf{e}^{(i)}$-direction
$\mathbf{y}_i$	vector of all fluid flow information concerning particle i; eqn. (2.16)
$\mathbf{Y}$	collection of particle information; $\mathbf{Y} := \{ \, \mathbf{y}_i \mid i \in \{1; ...; N\} \, \}$
$\mathcal{F}(\Omega(t))$	set of all measurable densities defined on $\Omega(t)$
$\delta_{\omega_N}(t)$	discrete measure defined on a particle ensemble $\omega_N(t)$
η	viscosity of the fluid; definition 2.4
Φ	ensemble of conservation variables; for one phase fluid flow $\Phi := (\rho\, , \, \rho v^{(1)}\, , \, \rho v^{(2)}\, , \, \rho v^{(3)}\, , \, \rho E)$
λ	heat conduction coefficient of the fluid; definition 2.4
∇	differential operator with respect to space variable $\mathbf{x}$ or $\mathbf{y}$; $\nabla := (\, \frac{\partial}{\partial x^{(1)}} \, \frac{\partial}{\partial x^{(2)}} \, \frac{\partial}{\partial x^{(3)}} \,)$ or $\nabla := (\, \frac{\partial}{\partial y^{(1)}} \, \frac{\partial}{\partial y^{(2)}} \, \frac{\partial}{\partial y^{(3)}} \,)$
ν	number of considered dimensions; $\nu \in \{1; 2; 3\}$
$\Omega(t)$	flow domain; $\Omega(t) \subset \mathbf{R}^{\nu}$
$\bar{\Omega}$	$\bar{\Omega} := \{(t, \mathbf{x}) \vert t \in T, \, \mathbf{x} \in \Omega(t)\}$, definition 1.1.
$\omega_N(t)$	particle ensemble consisting of N particles; definition 1.1
Π	smoothed particle operator for classical SPH; definition 1.10
Π^n	smoothed particle operator for general SPH; definition 3.2
ρ	fluid density

Bibliography

[1] D.S. Balsara. *von Neumann Stability Analysis of Smoothed Particle Hydrodynamics - Suggestion of Optimal Algorithms*, J. Comp. Phy., 121, 357-372 (1995)

[2] W. Benz. *Smoothed Particle Hydrodynamics: A rieview*, NATO workshop, Les Arcs, France, page 269, 1989

[3] F. Bosnjakovic, K.F. Knoche. *Technische Thermodynamik*, überarbeitete Auflage, Steinkopff Verlag, Darmstadt, 1988

[4] C.V. Chaubal, A. Srinivasan, Ö. Egecioglu, L.G. Leal. *Smoothed particle hydrodynamics techniques*, J. Non-Newtonian Fluid Mech., 70 (1997) 125-154

[5] H. Deconinck. *Upwind Methods and Multidimensional Splittings for the Euler Equations*, Preprint 1991-08/AR, van Karman Institute for Fluid Dynamics, Rhode Saint Genese, Belgium, 1991

[6] P. Deuflhard, F. Bornemann. *Integration gewöhnlicher Differentialgleichungen*, de Gruyter, Berlin, 1994

[7] G.A. Dilts. *Moving Least Squares Particle Hydrodynamics I, Consitency and Stability*, Hydrodynamic Methods Group, Los Alamos National Laboratory, 1996

[8] M.S. Fulbright, W. Benz, M.B. Davies. *A Method of Smoothed Particle Hydrodynamics Using Spheroidal Kernels*, The Astrophysical Journal, 440:254-262, 1995

[9] R.A. Gingold, J.J. Monaghan. *Shock Simulation by Particle Method SPH*, J. Com. Phy., 52, 374-389 (1983)

[10] R.A. Gingold, J.J. Monaghan. *Kernel Estimates as a Basis for Genreal Particle Methods in Hydrodynamics*, J.Com.Phy., 46,429-453(1982)

[11] S. Hiermaier. *Numerische Simulation von Impaktvorgängen mit einer netzfreien Lagrangemethode (Smoothed Particle Hydrodynamics)* , Diss., Mitteilungen

des Instituts für Mechanik und Statik 8, Universität der Bundeswehr, München, 1996

[12] C. Hirsch. *Numerical Compuation of Internal and External Flows*, John Wiley & Sons Ltd., 1988

[13] M. Hartig. *Randbedingungen für Partikelmethoden in der Strömungsmechanik*, Diplomarbeit, Fachbereich Mathematik, Universität Kaiserslautern, 1996

[14] K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching*, Springer-Verlag, 1984

[15] B.Manoj Kumar, S.V.Raghurama Rao, S.M. Deshpande. *A Kinetic Smooth Particle Hydrodynamics Method Using Peculiar Velocity Based Upwinding and Least Squares*, Report 98 FM 2, Centre of Exellence in Aerospace CFD, Indian Institute of Science, Bangalore, 1998

[16] R. Di Lisio, E. Grenier, M. Pulvirenti. *The Convergence of the SPH Method*, Computers Math. Applic. Vol 35, No. 1/2, pp. 95-102, 1998

[17] J. Monaghan. *Smoothed Particle Hydrodynamics*, Annu. Rev. Astronom. Astrophys. 30, 543 (1992)

[18] J. Monaghan. *Simulating free surface flows with SPH*, J. Com. Phy., 109,67-93 (1993)

[19] J.P. Morris. *An Overview of the Method of Smoothed Particle Hydrodynamics*, Berichte der AGTM, 95-152, Fachbereich Mathematik, Universität Kaiserslautern, 1995

[20] H. Neunzert, A. KLar, J. Struckmeier. *Particle Methods: Theory and Applications*, Berichte der AGTM, Bericht 95-153, Fachbereich Mathematik, Universität Kaiserslautern, 1995

[21] H. Neunzert. *Particle Methods*, Vorlesungsskript, Universität Kaiserslautern, 1991

[22] J. Radtke. *Numerical Simulation of Paper-Fluid Interaction*, Diplomarbeit, Fachbereich Technomathematik, Universität Kaiserslautern, 1999

[23] V. Ramesh, A.K. Ghosh, S.M. Deshpande, *Computation of Three Dimensional Inviscid Compressible Flows Using Least Squares Kinetic Upwind*

Method (LSKUM), Report 97 FM 8, Centre of Exellence in Aerospace CFD, Indian Institute of Science, Bangalore, 1997

[24] Y. Saad. *Iterative Methods for Sparse Linear Systems*, PWS Publ., Boston, Mass., 1996

[25] U. Scheffer, S. Hiermaier. *Improving an SPH code by alternative interpolation schemes*, 7.Fachtagung Baustatik/Baupraxis, Aachen, 18./19.3.1999

[26] H.R. Schwarz. *Numerische Mathematik*, Teubner Verlag, Stuttgart, 1986

[27] H. Schwetlick, H. Kretzschmar. *Numerische Verfahren für Naturwissenschaftler und Ingenieure*, Fachbuchverlag, Leipzig, 1991

[28] J. Stoer, R. Bulirsch. *Numerische Mathematik*, Springer-Verlag, 1990

[29] J.W. Swegle, D.L. Hicks, S.W. Attaway. *Smoothed Particle Hydrodynamics Stability Analysis*, J. Comp. Phy., 116, 123-134 (1995)

[30] E. Truckenbrodt. *Strömungsmechanik, Grundlagen und technische Anwendungen*, Springer-Verlag, 1968

[31] J.P. Vila. *On Particle Weighted Methods and Smooth Particle Hydrodynamics*, preprint, Institut National des Sciences Appliquees de Touloule, France, 1994

[32] H. Yserentant. *Particle Methods*, preprint, Mathematisches Institut der Universität Tübingen, Germany, 1994

[33] M. Schäfer, S. Turek. *Benchmark Computations of Laminar Flow Around a Cylinder*, in 'Notes on Numerical Fluid Mechanics', volume 52, ed. E.H. Hirschel, Vieweg & Sohn Verlagsges., Wiesbaden, 1996

Appendix: Some Computational Results

In the appendix we would like to exhibit some computational results. These will be

- a single shock with in-viscid fluid (section A.1.)
- a flow through a convergent tunnel with $Ma > 1$, in-viscid flow (section A.2.)
- a moving stiff plate in a box (moving boundary), in-viscid flow (section A.3.)
- a moving boundary problem where results are compared to the results of an FVM with moving grid, in-viscid flow (section A.4.)
- a viscous flow between two rotating cylinders (section A.5.)
- a viscous flow around a cylinder (section A.6.)
- an airbag related problem (section A.7.)

A.1. A Single Shock Solution with Inviscid Flow

The single shock experiment is a very easy one. As flow domain, we employ a box of length 20 and width 6. The sketch below represents a snapshot of the particle distribution at some time t.

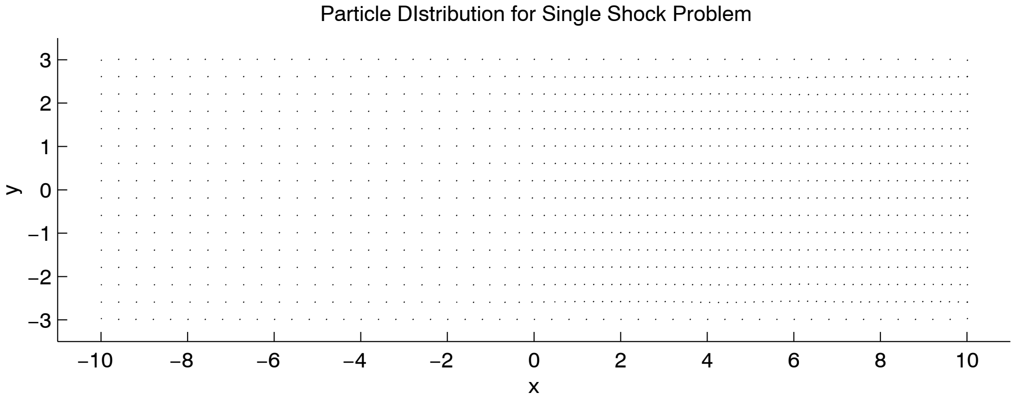


Fig. A.1: Particle distribution for single shock problem

At the left end, there is an inflow boundary, at the right end there is an outflow boundary, the upper and lower bounds are solid walls. The inflow and outflow conditions are chosen such that a stationary shock will evolve in the tube. To obtain the stationary solution, we continued to perform time steps until the solution did not change significantly. This example is chosen to show that general SPH is able to compute shocks very accurately.

The smoothing length is $h = 1$. The fluid employed is air. The flow is Euler.

We exhibit three results for shocks of different strengths with the first result being for the 'strongest' shock and the last result for the 'lightest' shock.

Besides the pressure and the velocity, we exhibit the solution for the physical entropy. As one notices, the numerical solution hits the right entropy production of the shocks. Computing the entropy production correctly is a nice indication for the fact, that the numerical scheme meets indeed some continuous conservation properties as exhibited in section 2.3.

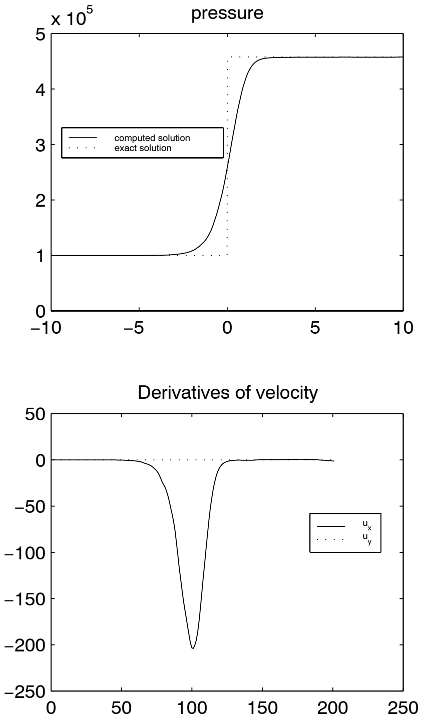


Fig. A.2: Solution 1 for high ratio of upwind and downwind pressure

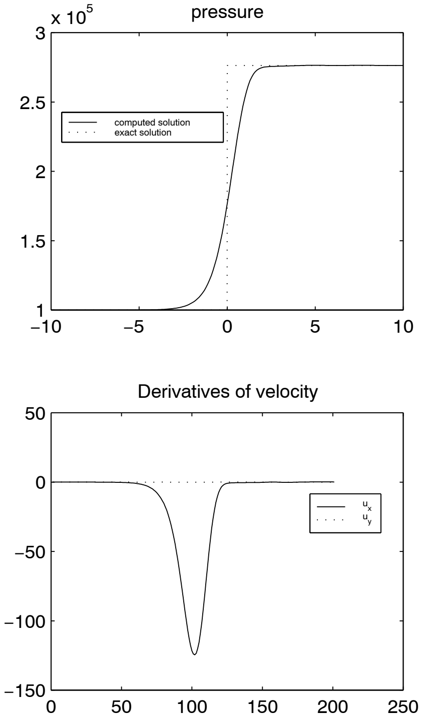


Fig. A.3: Solution 2 for medium ratio for upwind and downwind pressure

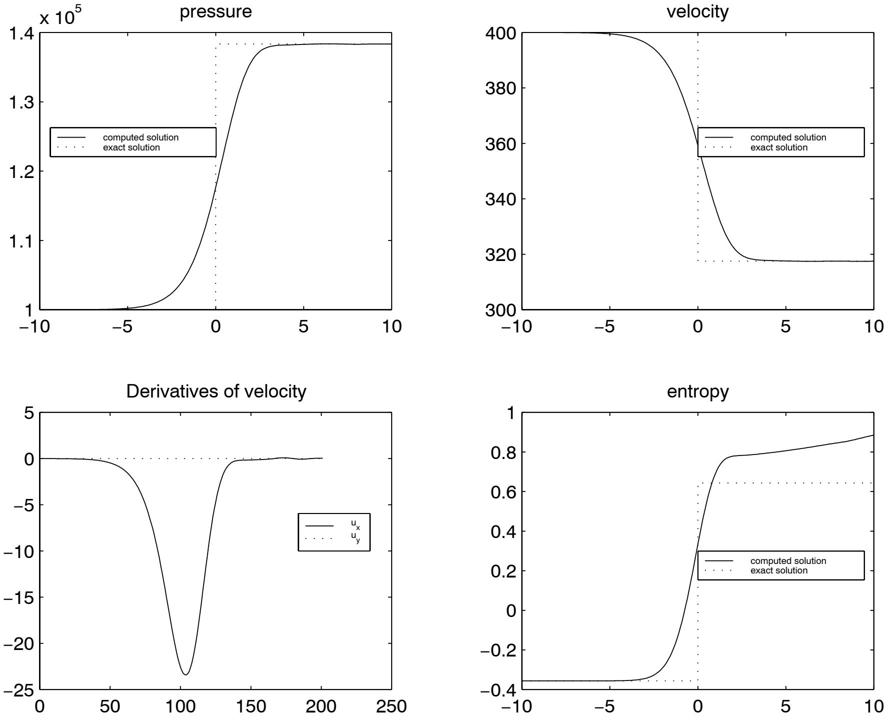


Fig. A.4: Solution 3 for low ratio of upwind and downwind pressure

We observe that shocks are reproduced very accurately. The width of a shock is approximately $4 \cdot h$. For light shocks, the entropy production is a too large, and also the computed width of the shock is larger than for stronger shocks. Up to now, we did not find any explanation for this phenomenon.

A.2. A Shock System in a Convergent Channel

The example of a shock system in a convergent channel is chosen to demonstrate that not only 1D-shocks, but also shocks in more dimensions are easily computed with the general SPH method. As flow domain we choose a convergent channel as demonstrated below.

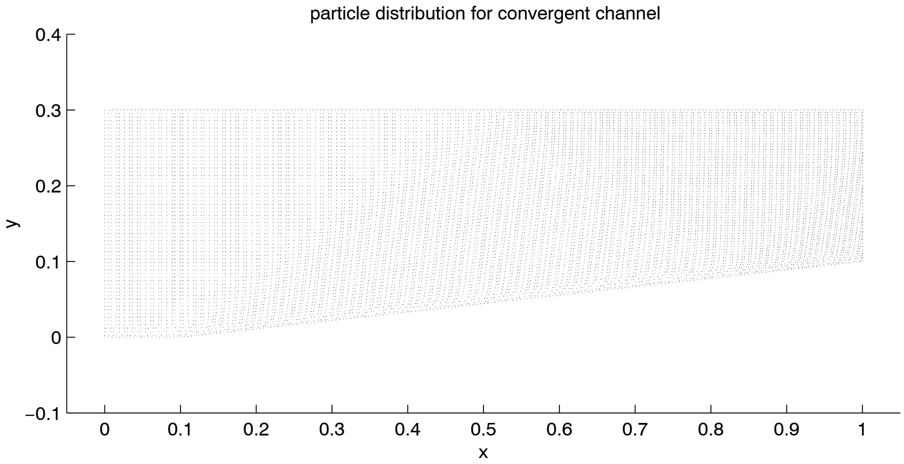


Fig. A.5: Geometry and particle distribution for convergent channel

At the left end there is supersonic inflow, at the right end there is outflow, the upper and lower bounds are solid walls. At the lower wall, at $x^{(1)} = 0.1$ there is a bend. As inflow conditions we chose
$p_0 = 1.0\ bar$
$T_0 = 300.0\ K$
$v_0^{(1)} = 700 \frac{m}{s}$
$Ma_0 = 2.0$
The smoothing length was chosen as $h = 0.01$. The fluid employed is air, the flow is in-viscid. We performed as much time steps as we needed to achieve a stationary solution.

Since the inflow is supersonic, we can easily compute an analytic solution to that problem and compare the numerical result with the analytic one. The following two plots show the numerical result and the analytic solution.

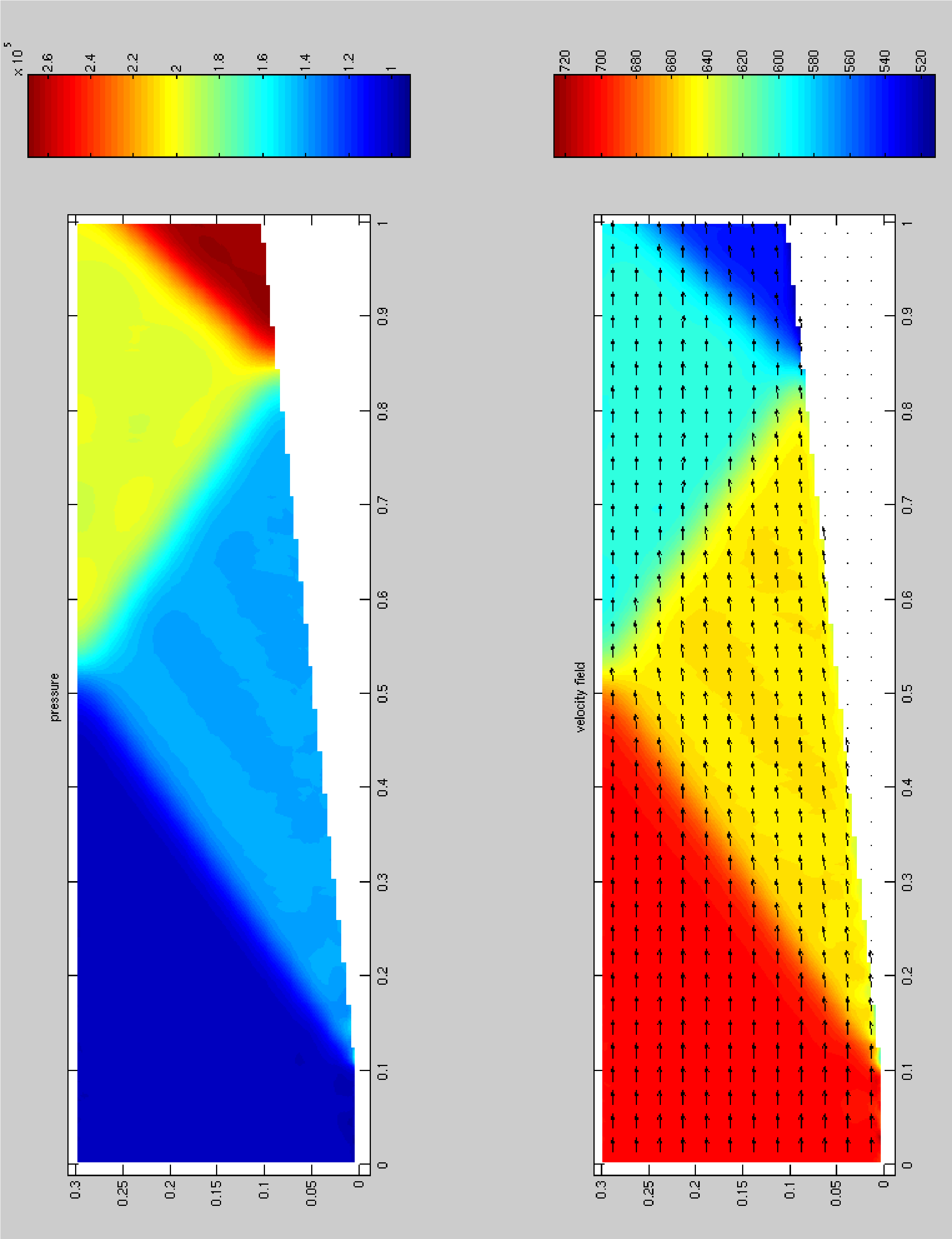

pressure
velocity field
x 10^5
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
720
700
680
660
640
620
600
580
560
540
520
0
0.05
0.1
0.15
0.2
0.25
0.3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fig. A.6: Numerical solution to the convergent channel problem

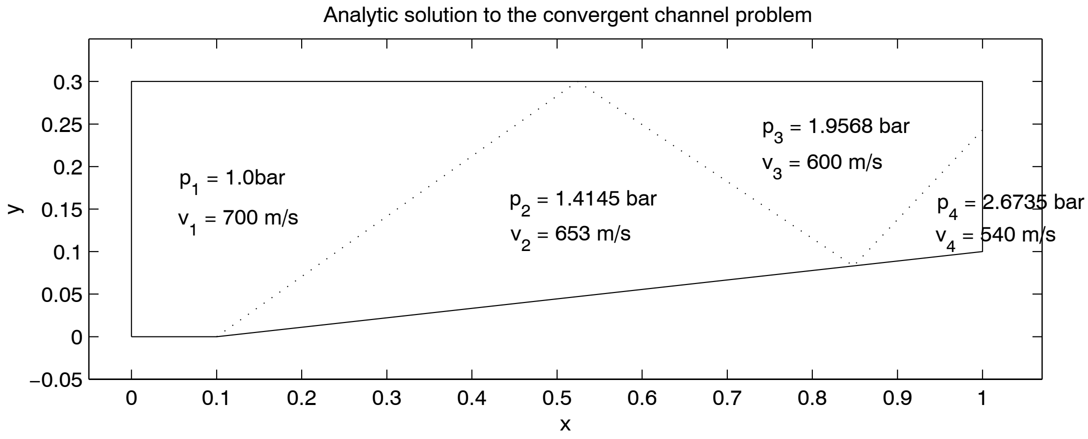


Fig. A.7: Analytic solution for the convergent channel problem

The comparison between the two solutions shows, that the positions of the shocks are reproduced very accurately.

A.3. Moving Stiff Plate in a Closed Box

This example gives an impression on how the general SPH works for problems with sharp edges. The motivation for a good treatment of sharp edges is given by some problems related to fluttering paper sheets (which is not an academic, but a real industrial problem). For this example we simply consider a stiff plate placed horizontally in a closed rectangular box. The plate moves downwards. A snapshot of the geometry and particle distribution at some time t is shown in the figure below.

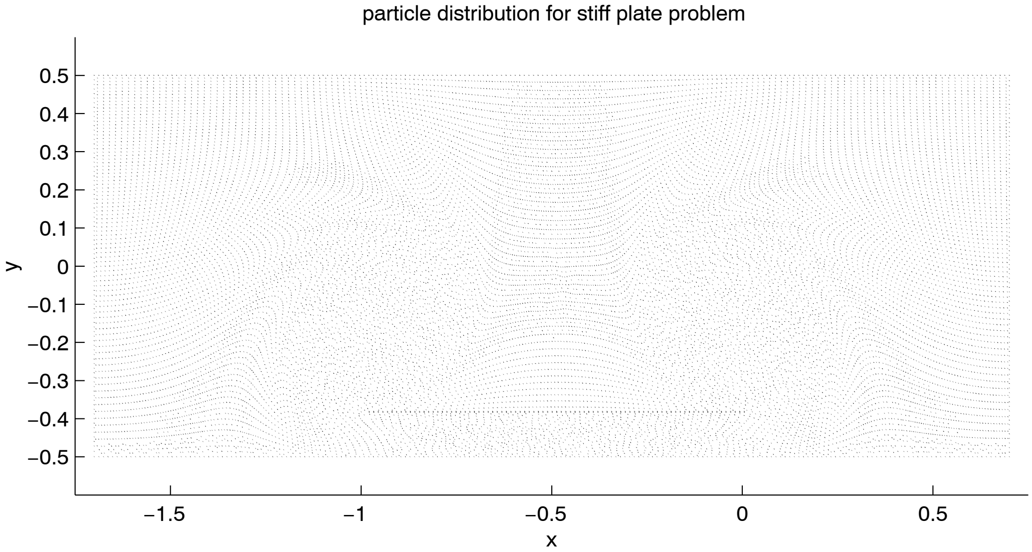


Fig. A.8: Particle distribution for the stiff plate problem

The stiff plate has zero thickness, it is represented by the set of points

$$\mathcal{S}_{sp}(t) := \{ \ (x^{(1)}, x^{(2)}) \mid -1 \leq x^{(1)} \leq 0 \ ; \ x^{(2)} = -\frac{1}{2}\, 3000\frac{m}{s^2} \cdot (t - t_0)^2 \ \}$$

i.e. at the initial time t_0 the plate is placed at $x^{(2)} = 0$ with zero velocity and is then moved downwards with a constant acceleration of $3000\frac{m}{s^2}$. The particle distribution in figure A.8 is a snapshot where the plate is at about $x^{(2)} = -0.36$ as the reader may notice. The used fluid is air. At the initial time t_0, the velocity of the fluid is zero, $p = 1bar$, and $T = 300K$. The flow is assumed to be in-viscid. As usual, we assume slipping condition at all solid walls and at the surface of the plate.

This example provides a big advantage. As the stiff plate approaches the lower wall, we can write down an analytic solution for the fluid flow between the stiff plate and the wall, under the assumption that

- the flow is incompressible

- $v^{(2)}$ is a linear function (which is approximately the case as the gap between lower wall and plate becomes thin)

For these assumptions, we observe that $\frac{\partial v^{(1)}}{\partial x^{(1)}} \gg \frac{\partial v^{(1)}}{\partial x^{(2)}}$ and $\frac{\partial p}{\partial x^{(1)}} \gg \frac{\partial p}{\partial x^{(2)}}$ and therefore we write down $v^{(1)}$ and p within the gap only as functions of $x^{(1)}$. The obtained analytic solutions are

$$v^{(1)} = \frac{|v|}{h_0} \cdot (x^{(1)} + 0.5)$$

$$p = \rho \frac{|v|^2}{h_0^2}(0.5 - x^{(1)})(0.5 + x^{(1)}) + p_0$$

where $|v|$ is the velocity of the moving plate and h is the width of the gap. The two following plots show snapshots of the flow fields for two different positions of the stiff plate (first $x^{(2)} = -0.105$, second $x^{(2)} = -0.36$).

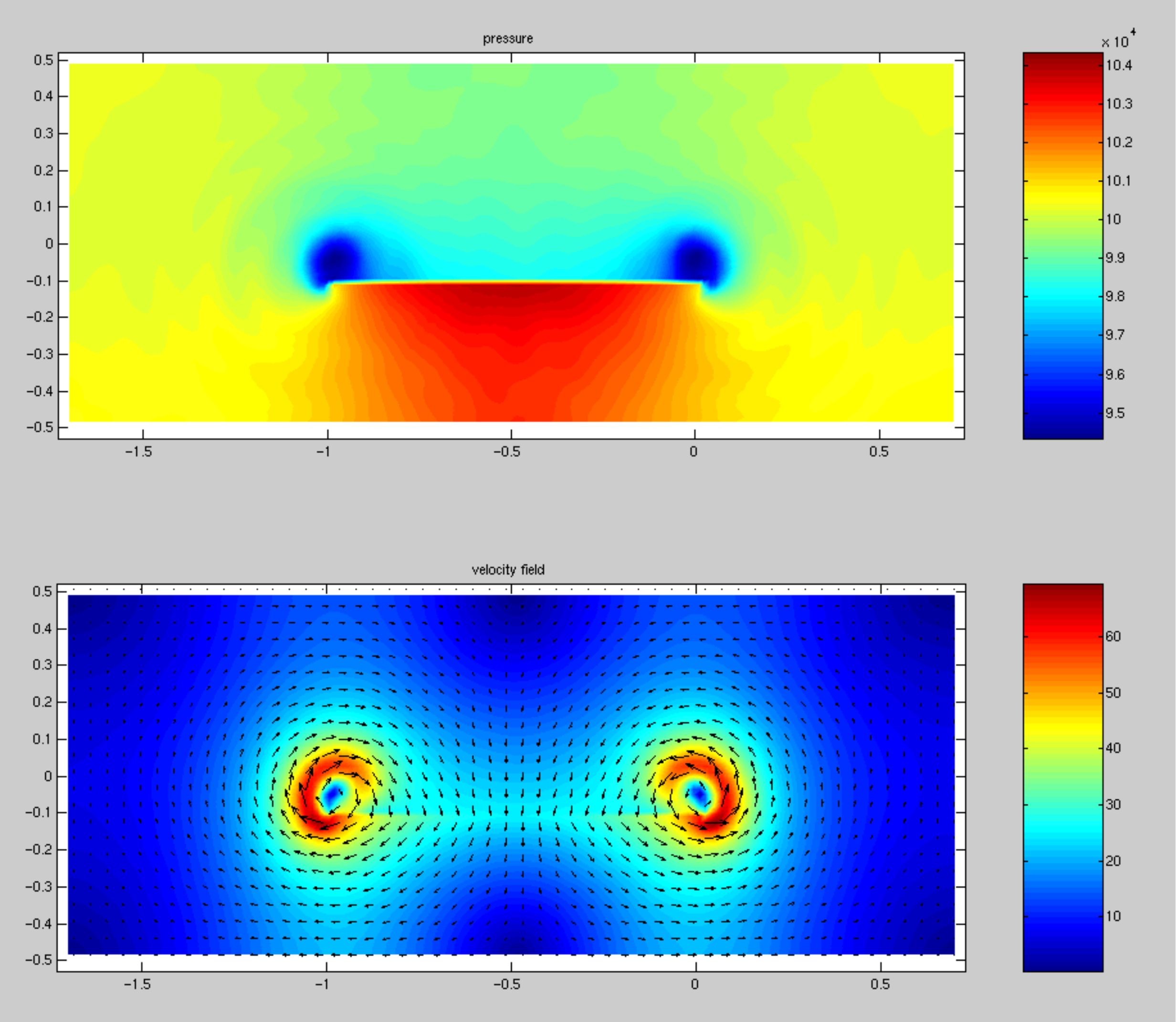


Fig. A.9: Numerical result for plate at $x^{(2)} = -0.105$

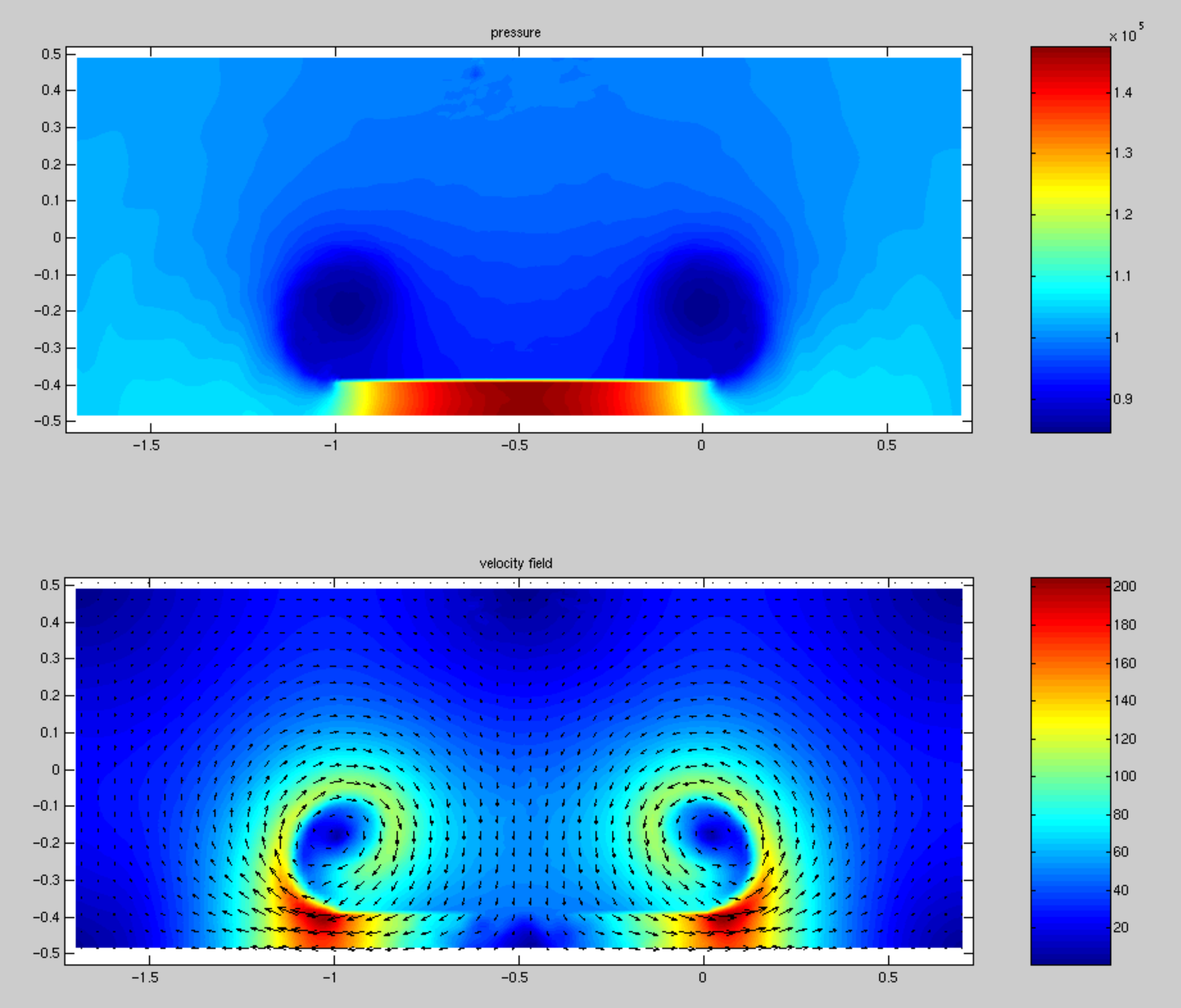


Fig. A.10: Numerical result for plate at $x^{(2)} = -0.36$

For plate position $x^{(2)} = -0.36$, the next plot compares the computed results for pressure and velocity in the gap with the analytic solutions. We have to take into account that the analytic solution is valid for incompressible fluids. However, the numerical computations are based on compressibility of the fluid. Thus, we have to mention that that Mach number of the flow at plate position $x^{(2)} = -0.36$ is $Ma \approx 0.3$. Nevertheless, the two solutions seem to be very well comparable.

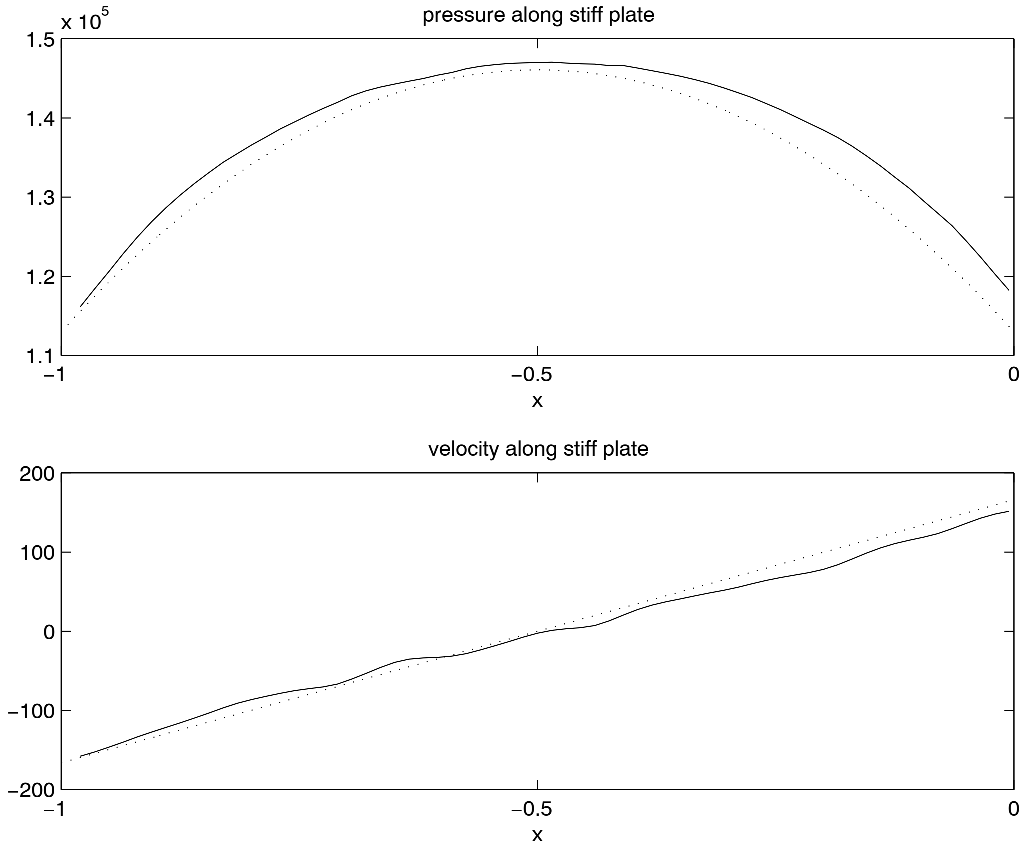


Fig. A.11: Comparison between numerical and analytic solutions at $x^{(2)} = -0.36$

The solid lines are the numerical and the dotted lines the analytic solutions.

A.4. 'Moving Nose' Problem

The 'moving nose' problem is a flow problem in a closed chamber which changes its geometry in time. We have straight solid wall boundaries at the left, right, and bottom which do not move. The upper boundary is cosine-shaped, forming the 'nose'. The 'nose' moves from left to right with constant velocity v_0. The following plot shows a snapshot of the geometry and the particle distribution of the 'moving nose' problem.

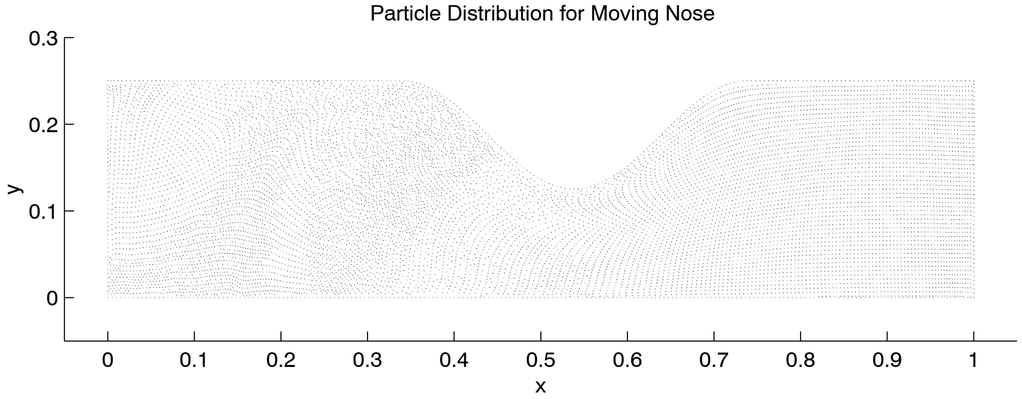


Fig. A.12: Geometry and particle distribution for 'moving nose' problem

The set of geometrical points of the upper boundary is given by

$$\mathcal{S}_{ub}(t) = \{\ (x^{(1)}, x^{(2)}) \mid 0 \leq x^{(1)} \leq 1\ ;\ x^{(2)} = f_{ub}(t, x^{(1)})\ \}$$

where the function $f_{ub}(t, x^{(1)})$ is defined as

$$f_{ub}(t, x^{(1)}) \ = \ \begin{cases} \frac{1}{4} - \frac{1}{8} \cos \left[\ 5\pi\ \left(x^{(1)} - (\frac{1}{5} + v_0(t - t_0))\right)\ \right] & \text{if } \mid x^{(1)} - (\frac{1}{5} + v_0(t - t_0)) \mid < \frac{1}{5} \\ \frac{1}{4} & \text{otherwise} \end{cases}$$

with $v_0 = 69.4\frac{m}{s}$. v_0 is obviously the speed which the cosine-shaped nose is moving with. It is chosen such that the speed of the nose is $Ma = 0.2$. The fluid is air. The computations are Euler with slipping condition at all wall boundaries. The smoothing length is $h = 0.01$. At the initial time t_0 the fluid velocity is zero, the pressure is 1 bar and the temperature is 300K throughout the flow domain.

For the same example, computations using a finite volume method were performed by J. Radtke (see [22]). His idea is to have a moving finite volume grid. In the following, solutions produced by general SPH and the moving grid FVM are compared for different times t. The FVM computations were tuned such that the number of FV-cells was about the number of particles used by the general SPH.

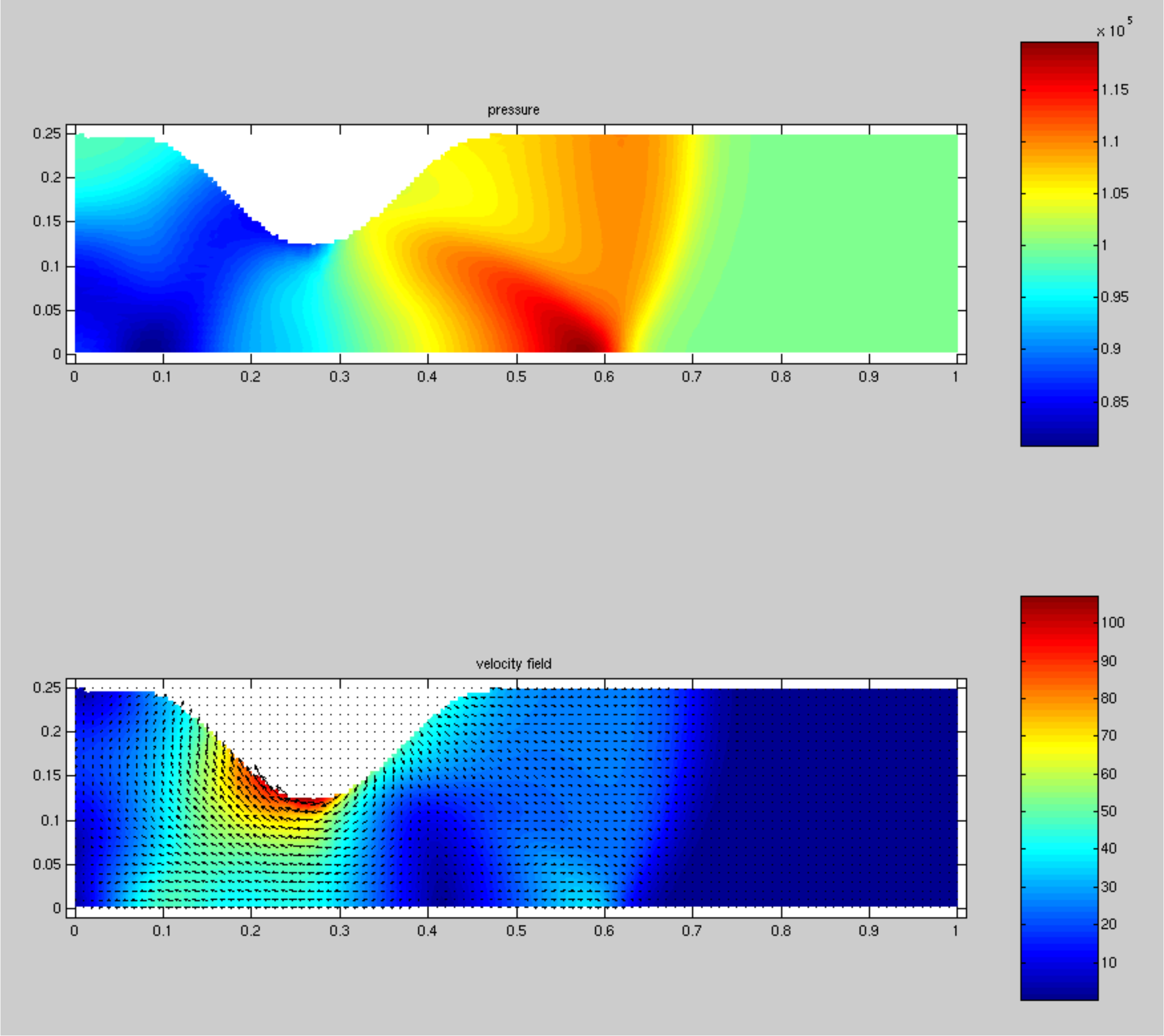


Fig. A.13: SPH solution to the moving nose problem for $t = 0.001s$

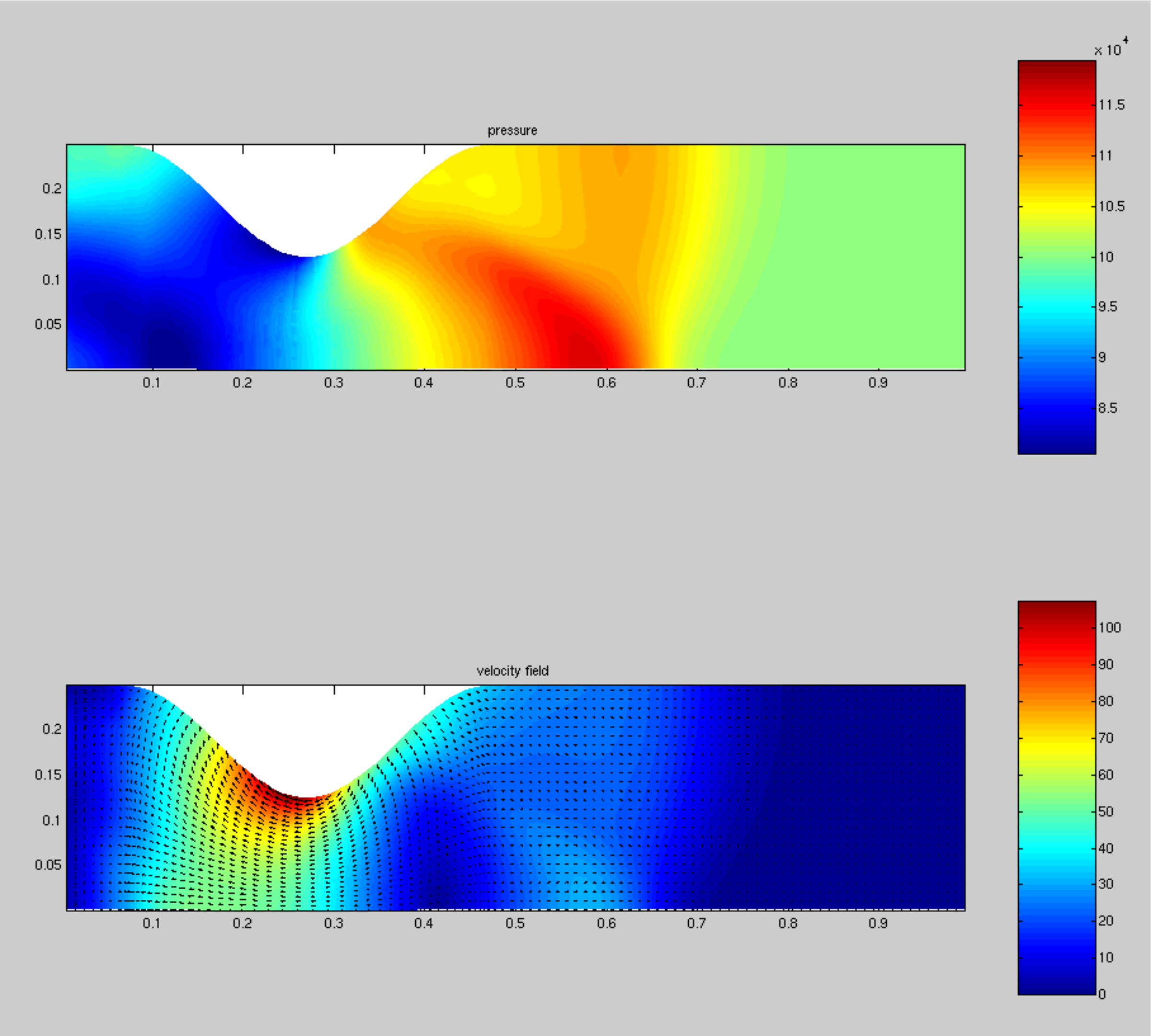


Fig. A.14: FVM solution to the moving nose problem for $t = 0.001s$

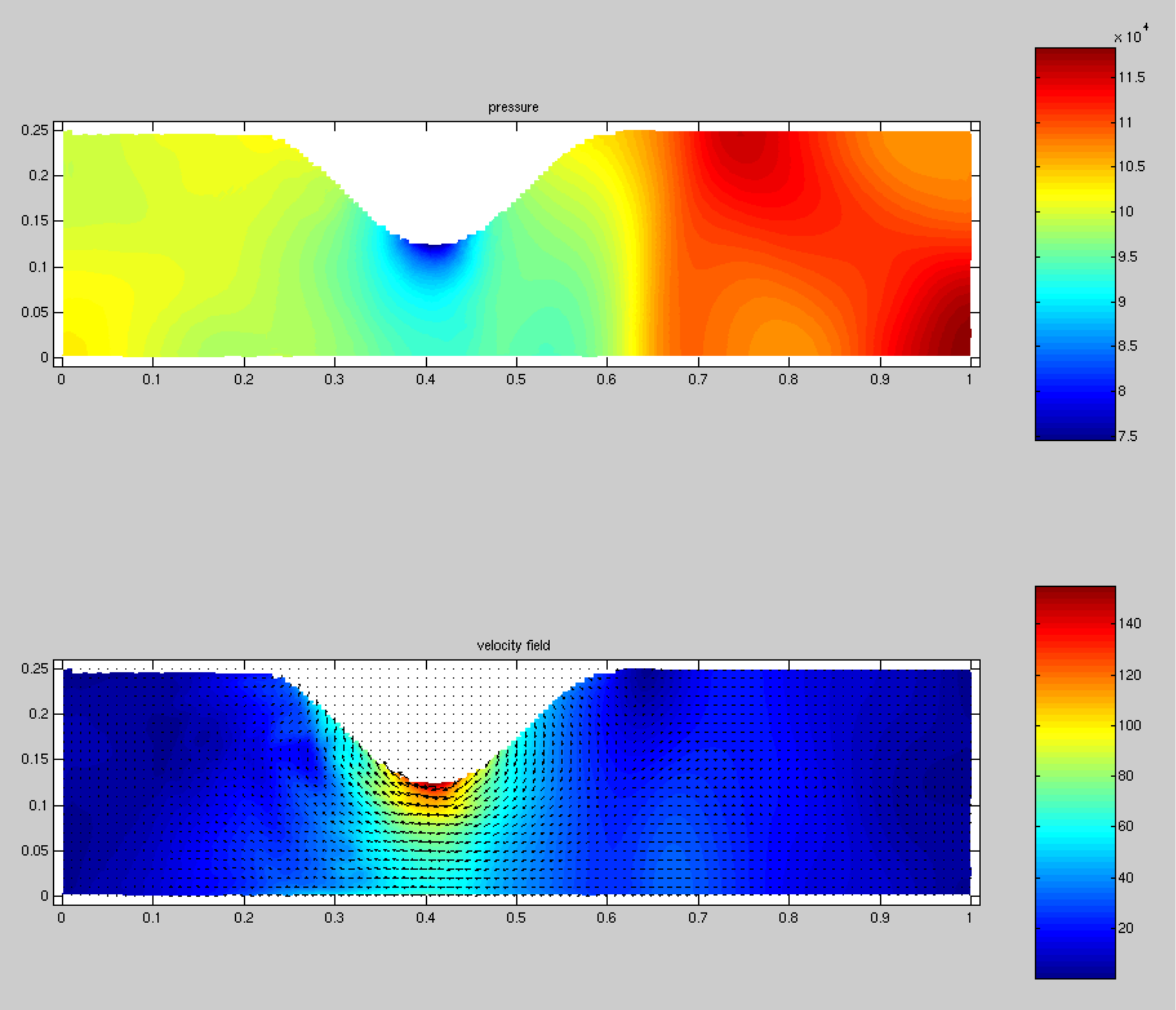


Fig. A.15: SPH solution to the moving nose problem for $t = 0.0035s$

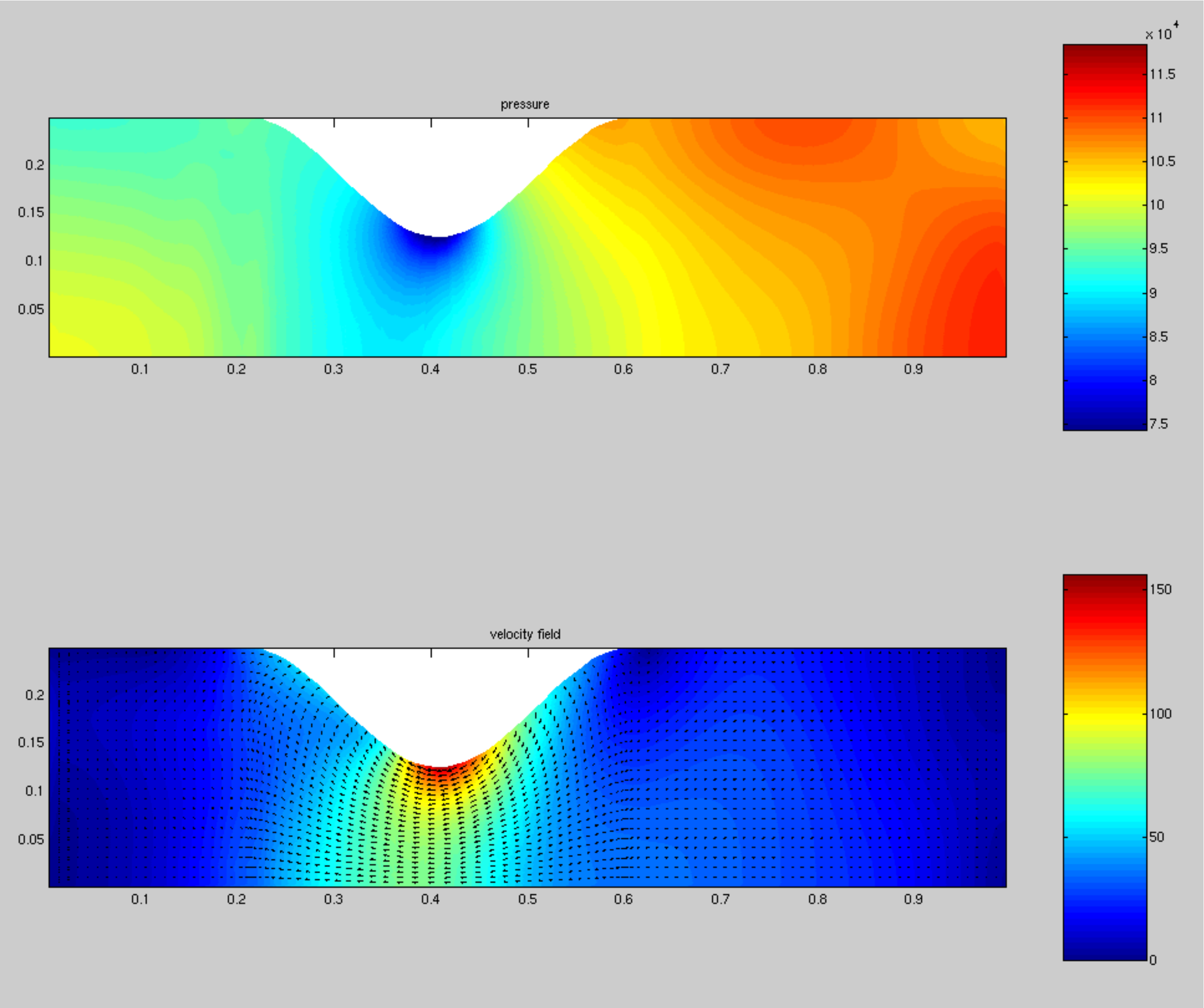


Fig. A.16: FVM solution to the moving nose problem for $t = 0.0035s$

A.5. Viscous Flow Between Two Rotating Cylinders

In example A.5. and A.6. we would like to exhibit results for viscous flow, demonstrating, that general SPH for Navier-Stokes equations works as well as for Euler equations. The problem considered in this section is a simple flow between two rotating cylinders. The outer cylinder has a radius of 1, the inner cylinder has a radius of 0.5. The outer cylinder rotates wit a constant velocity of $100\frac{m}{s}$, the inner one with $200\frac{m}{s}$. The following plot shows, as usual a snapshot of the geometry and the particle distribution of the problem.

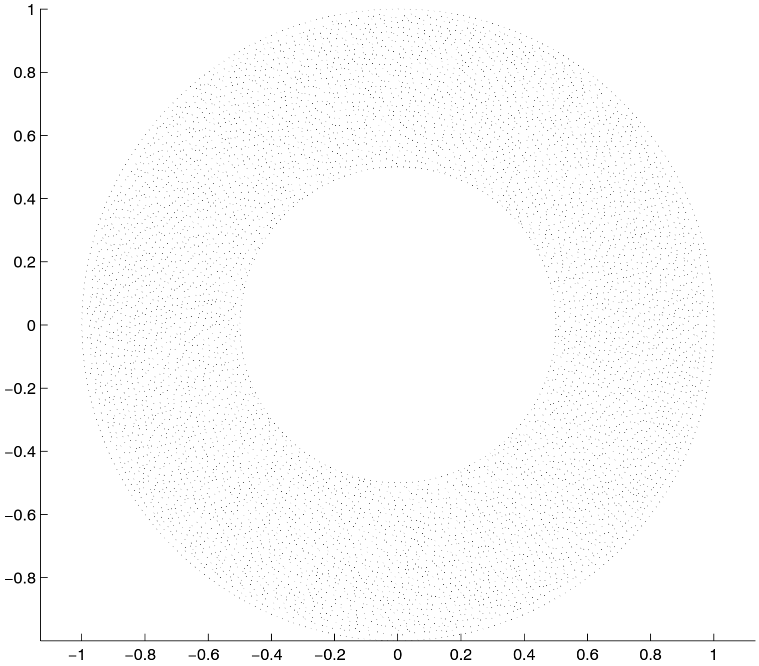


Fig. A.17 Geometry and particle distribution of the rotating cylinder problem

The used smoothing length is $h = 0.03$. The used fluid is air. The average density is $\bar{\rho} = 1.4\frac{kg}{m^3}$, the average temperature at t_0 was $\bar{T} = 300K$. We performed as many time steps as we needed to achieve a stationary solution. Under the assumption that the fluid is incompressible, we obtain an analytic solution to this problem, given by

$$\|\mathbf{v}\|_2 = c\,r^{-1}$$

$$p = p_0 - \frac{1}{2}\bar{\rho}\,c^2\,r^{-2}$$

where $c = 100\frac{1}{s}$ for this special case. The following plot shows the solution for the whole flow domain.

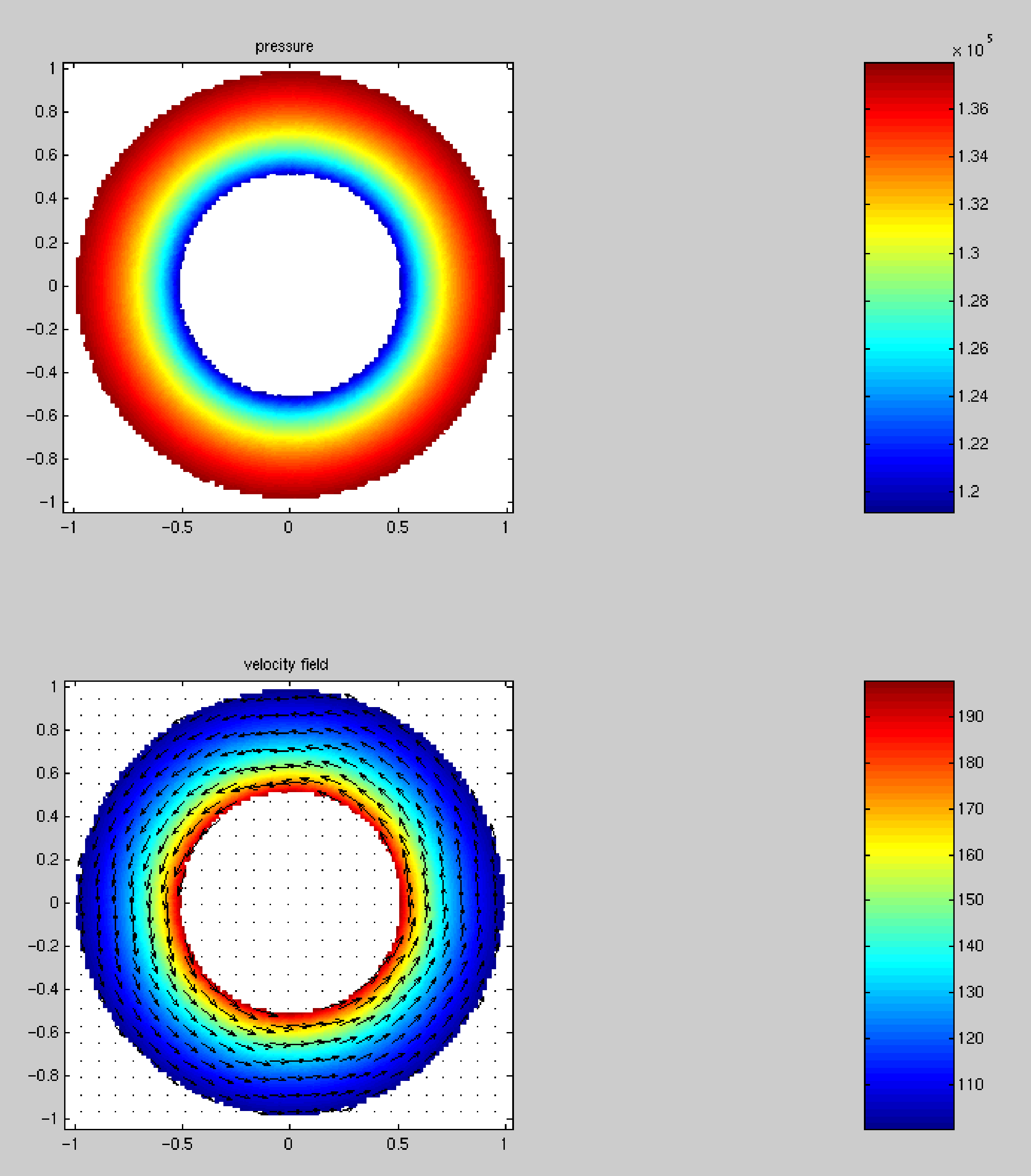
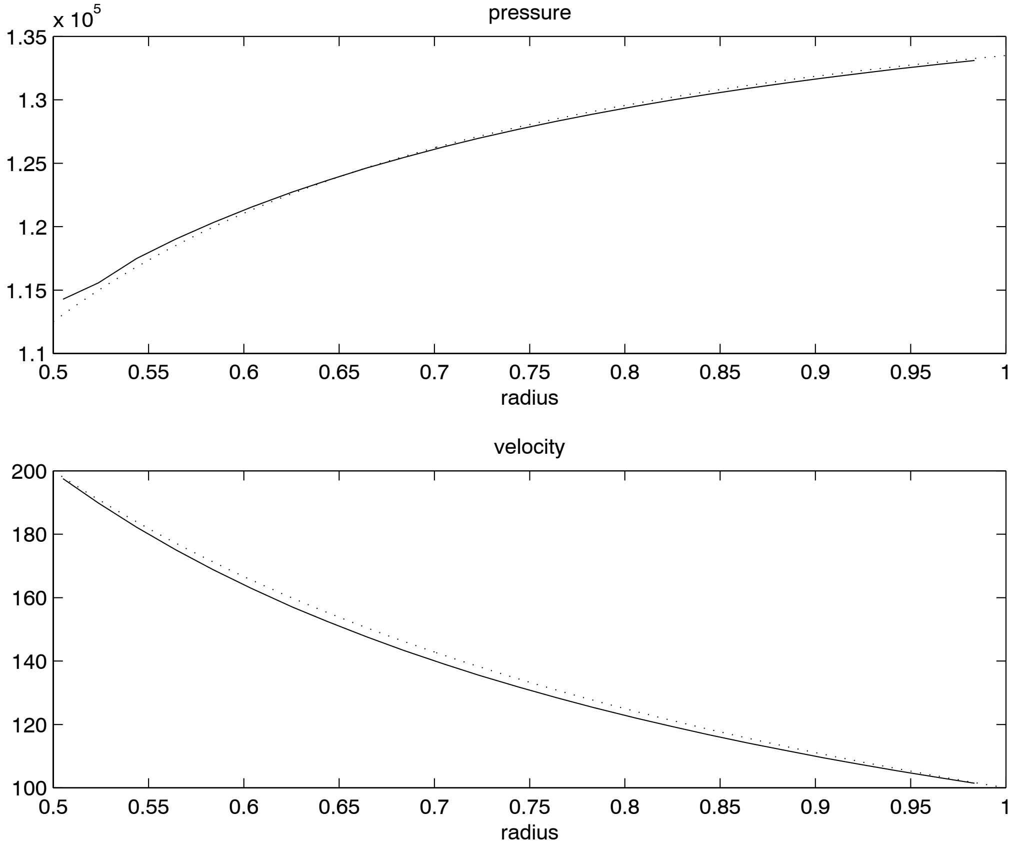


Fig. A.18 Numerical solution for the rotating cylinder problem.

The final plot gives a comparison between the numerical and the analytic solution. We will find the pressure and the velocity versus the radius.


pressure

velocity

Fig. A.19 Numerical and analytic solutions of p and $\|\mathbf{v}\|_2$ versus the radius.

A.6. Viscous Flow Around Cylinder

A classical example to verify a numerical method is the consideration of a viscous flow around a cylinder. This benchmark example is to be found in [33] and is shortly explained here. We discuss here the steady case ($Re = 20$). The flow domain is a rectangular box of length 2.2 and height 0.41. The lower left corner has the coordinates $(0, 0)$. In this channel there is a cylinder with diameter 0.1 whose center is at $(0.2, 0.2)$. The left boundary is an inflow boundary, the right boundary is an outflow boundary. The velocity at the inlet is given by

$$v^{(1)}(0, x^{(2)}) = 4U_m \, x^{(2)} \, (H - x^{(2)}) \, / \, H^2$$
$$v^{(2)}(0, x^{(2)}) = 0$$

The density is $\rho = 1.0 kg/m^3$. The kinematic viscosity is defined as $\nu = 10^{-3} m^2/s$. The flow is incompressible. Since the SPH idea is based on compressibility of the fluid, we chose a compressible fluid with gas constant R such, that the inflow Mach number was $Ma = 0.1$. The condition at the upper and lower solid walls and at the surface of the cylinder is no slip. The following plot shows the geometry and a snapshot of the particle positions at a certain time t.

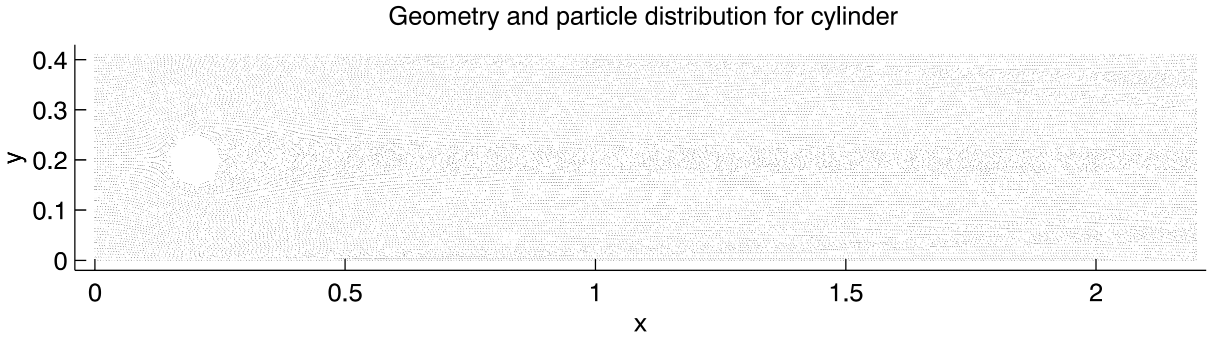


Fig. A.20 Geometry and particle distribution for the cylinder problem

For this problem there were carried out measurements. Measured results are available for drag coefficient c_D and the lift coefficient c_L about the cylinder, the length of the recirculation zone behind the cylinder L_a and the pressure difference between the front and the back point of the cylinder Δp. In the following we exhibit three result plots.

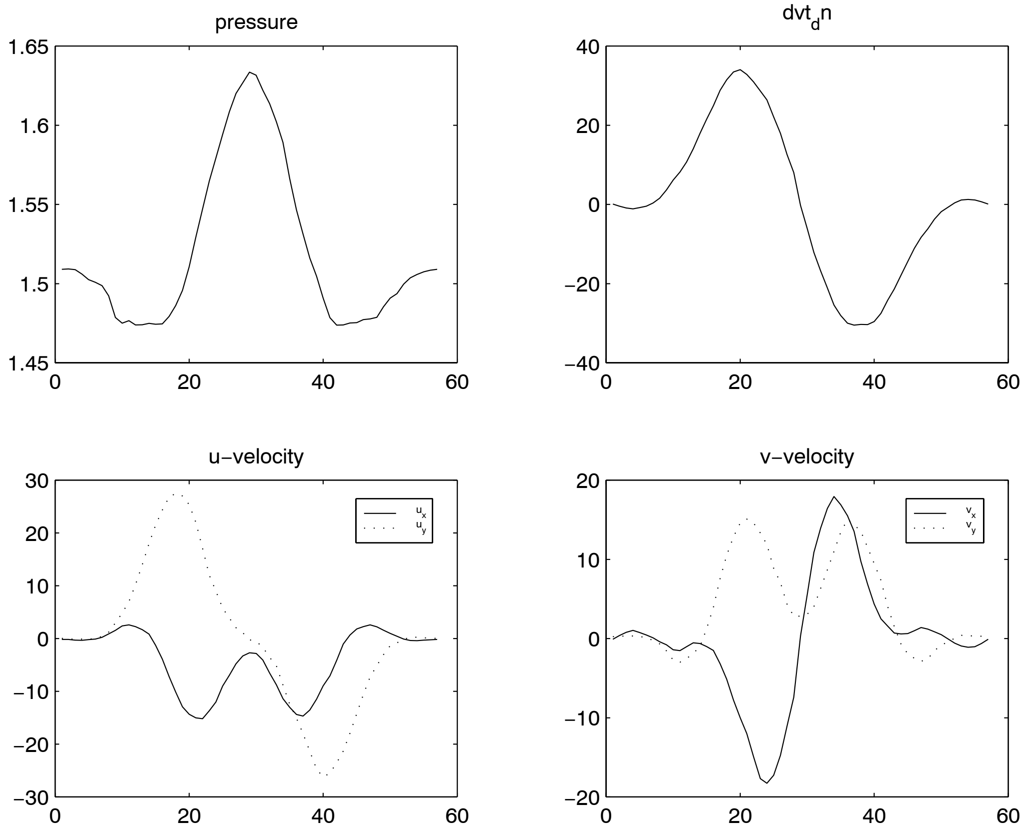


Fig. A.21 Some values as functions of surface variable

This plot gives the appropriate function values as one walks around the cylinclr in positive sense.

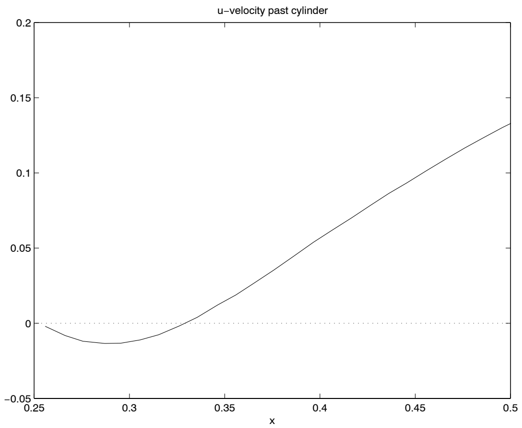


Fig. A.22 Flow velocity directly behind the cylinder

The two plot above allow us to give the values
$\Delta p \approx 0.115$; measured: lower bound 0.1172, upper bound 0.1176

$L_a \approx 0.08$; measured: lower bound 0.0842, upper bound 0.0852

The computed c_D and c_L are
$c_D \approx 5.58$; measured: lower bound 5.57, upper bound 5.59
$c_L \approx 0.05$; measured: lower bound 0.0104, upper bound 0.0110

The values above are to be understood as average values of the results at different times t. We notice that the lift coefficient is completely out of order.

Finally, we show the computed result in the whole flow domain.

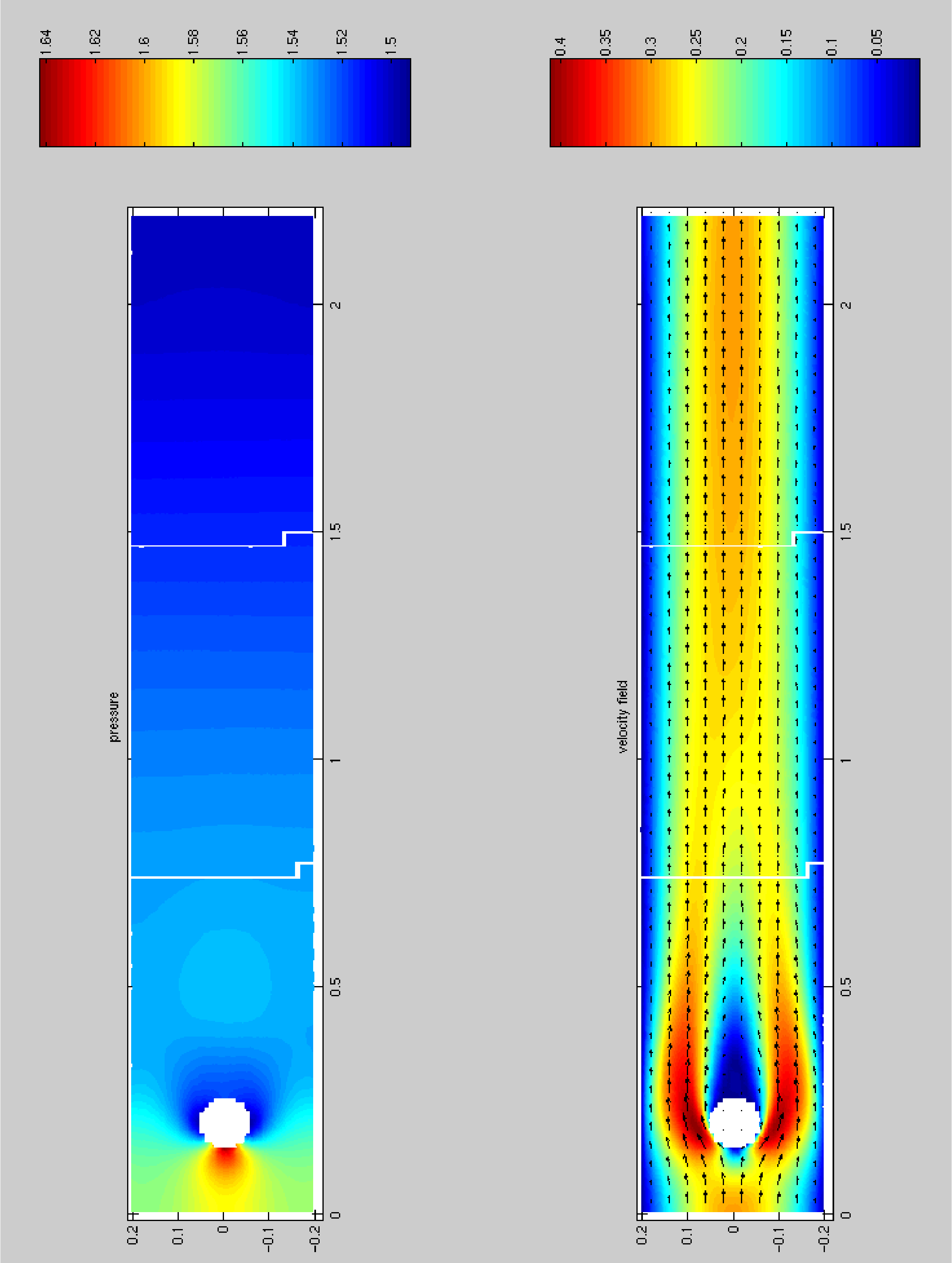


Fig. A.23 stationary flow of a viscous fluid about a cylinder

The white stripes represent the boundaries between the computation domains of the three different processors (this example was computed in parallel).

A.7. An Airbag Related Problem

Finally, we would like to turn to a problem with free surface. The problem presented here is a benchmark which has its roots in the car industry. Car producers are very much interested in simulating airbags blown up in the case of an car accident to protect the driver as well as passengers from being hit by solid parts of the car. The information one would like to extract from the simulation of the blown up airbag is, how well the airbag is able to protect the human being.

Since the airbag has a free boundary changing very rapidly in time, this problem is almost designed to be solved with a general SPH method.

Here, we would like to present a benchmark example, which is related to the airbag problem and which was brought up by a software company being in charge of simulating the blowing process of an airbag. This benchmark example was designed to compare the results of different numerical methods which should be able to compute the airbag. The comparison was not finished by the date of concluding this thesis, so there are only SPH results exhibited without any comparison.

First, let us describe the problem.

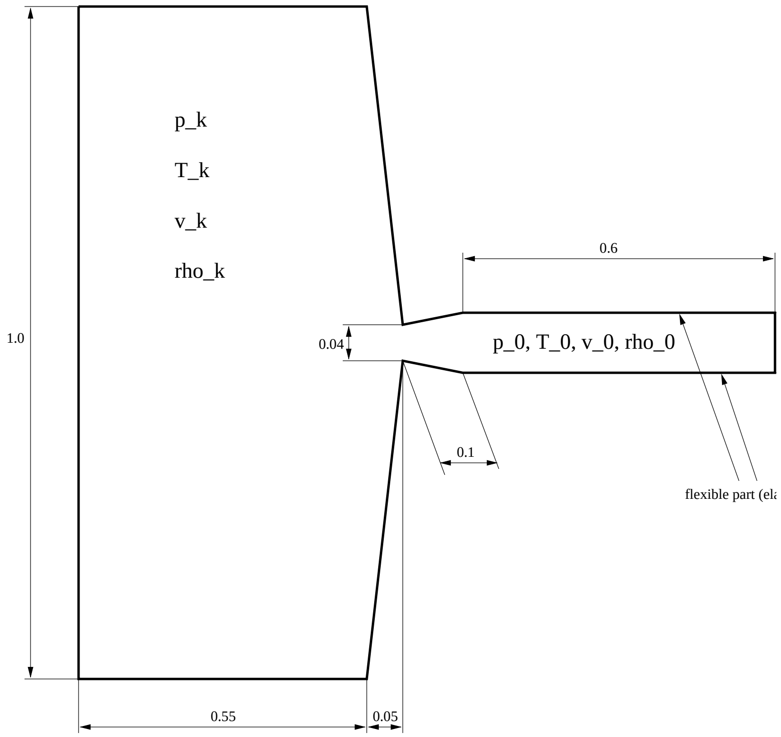


Fig. A.24 Geometry of the benchmark problem

Figure 24 shows the geometry of the example. The left part is a chamber, filled initially with air of $T_k = 1212\,K$, $p_k = 15.64 \cdot 10^5\,Pa$, $\mathbf{v}_k = \mathbf{0}\,m/s$ and $\rho_k = 4.72\,kg/m^3$. The boundaries of the chamber part are solid walls, which do not move. The right part is the flexible part. The marked horizontal lines are made of elastic material, having a density of $1848\,kg/m^3$, a thickness of $0.003\,m$ and an elastic modulus of $2 \cdot 10^8 Pa$. The state of the air in the flexible part at the initial point of time is $T_0 = 300\,K$, $p_0 = 1.0 \cdot 10^5\,Pa$, $\mathbf{v}_0 = \mathbf{0}\,m/s$ and $\rho_0 = 1.16\,kg/m^3$. At the initial time, the air in the chamber is allowed to flow and therewith to blow the flexible part. The numerical results for the flexible part are shown below for some times. At the end, we will find a plot, showing the particle at the final time. We shall see there, that this example was computed with variable smoothing length.

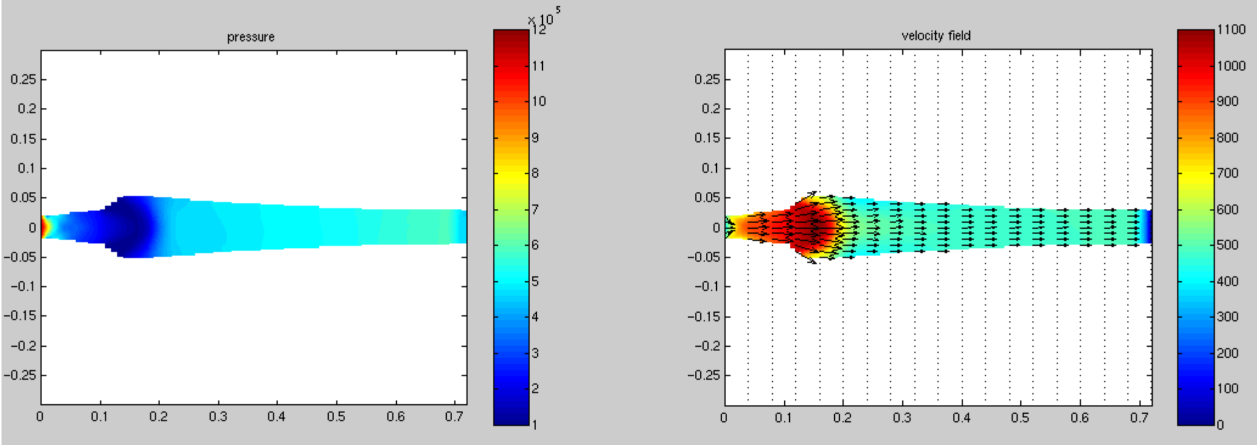


Fig. A.25 Velocity and pressure at time $t = 0.001s$

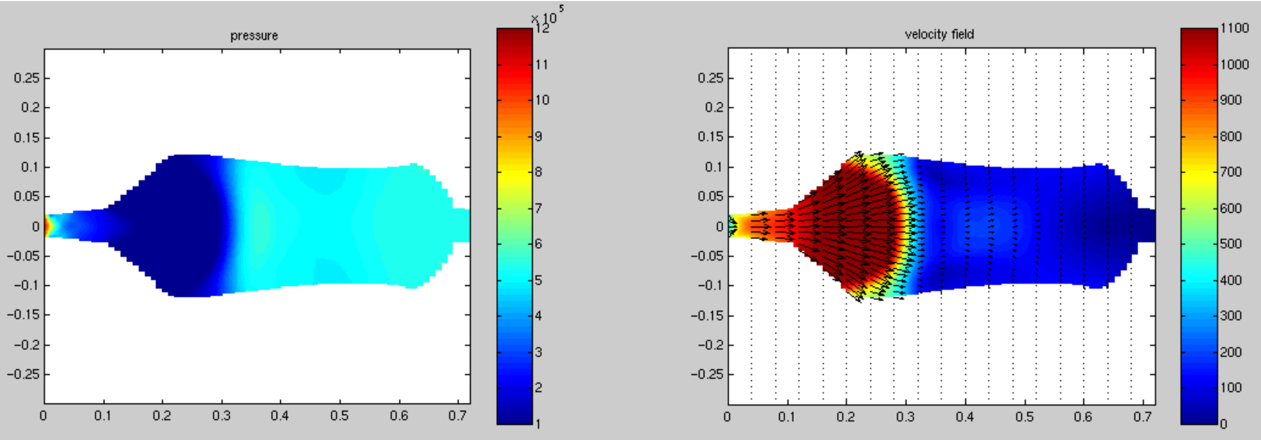


Fig. A.26 Velocity and pressure at time $t = 0.002s$

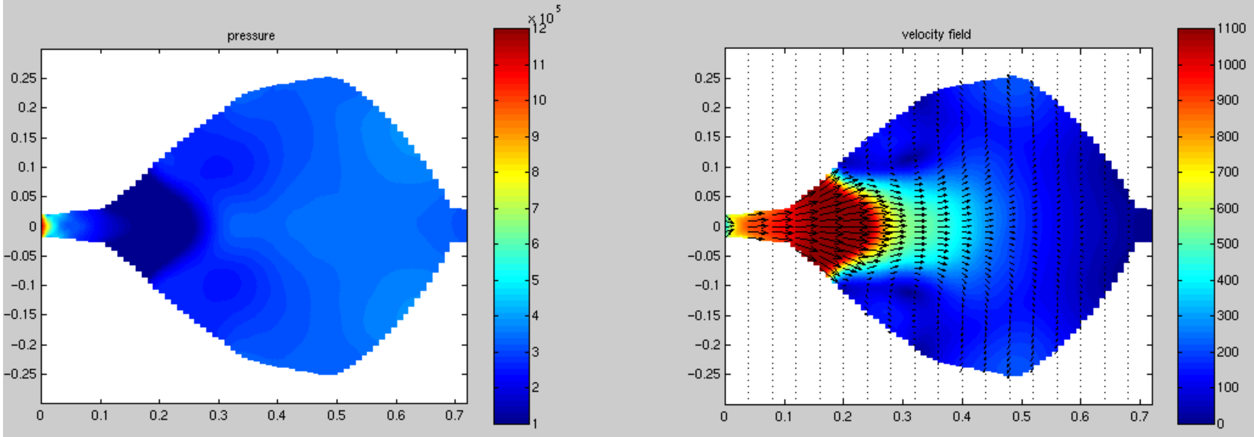


Fig. A.27 Velocity and pressure at time $t = 0.003s$

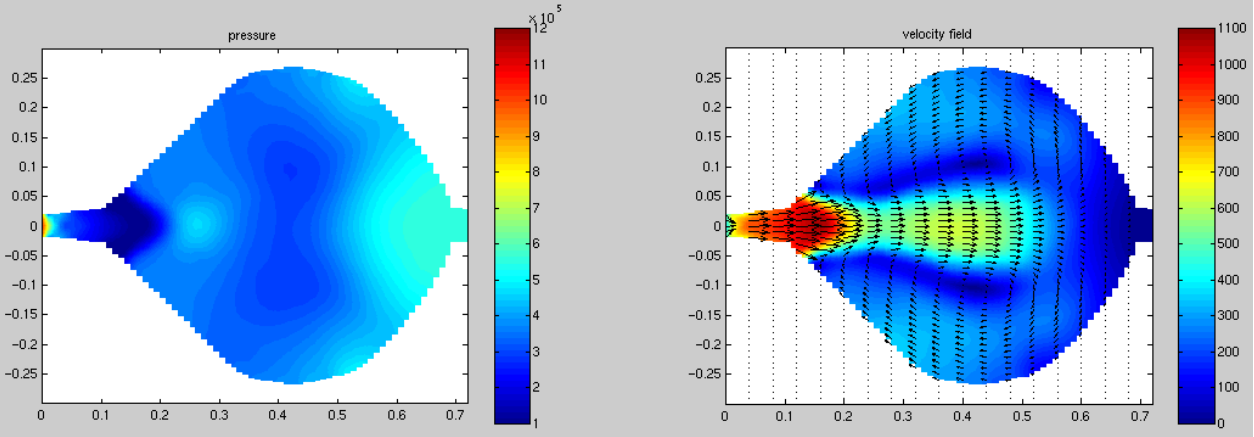


Fig. A.28 Velocity and pressure at time $t = 0.004s$

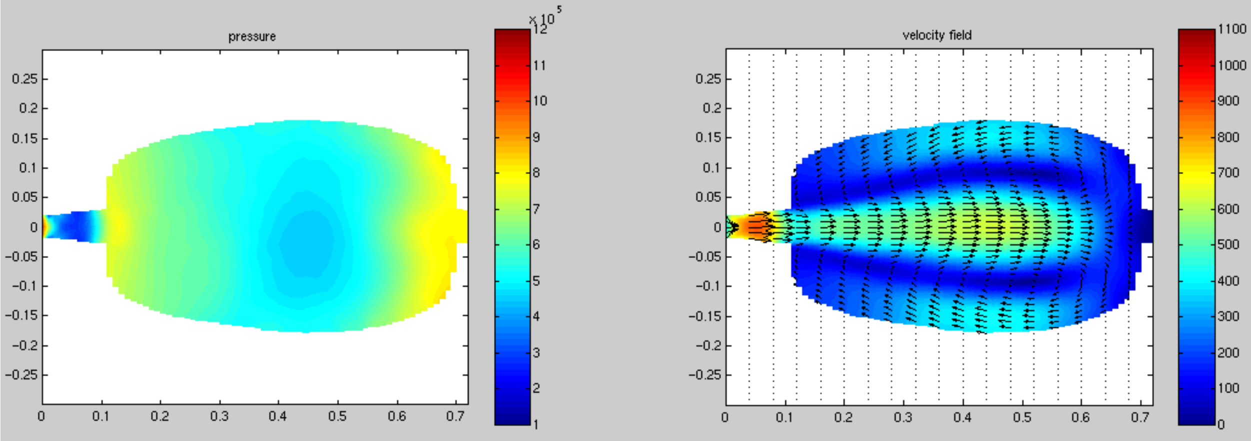


Fig. A.29 Velocity and pressure at time $t = 0.005s$

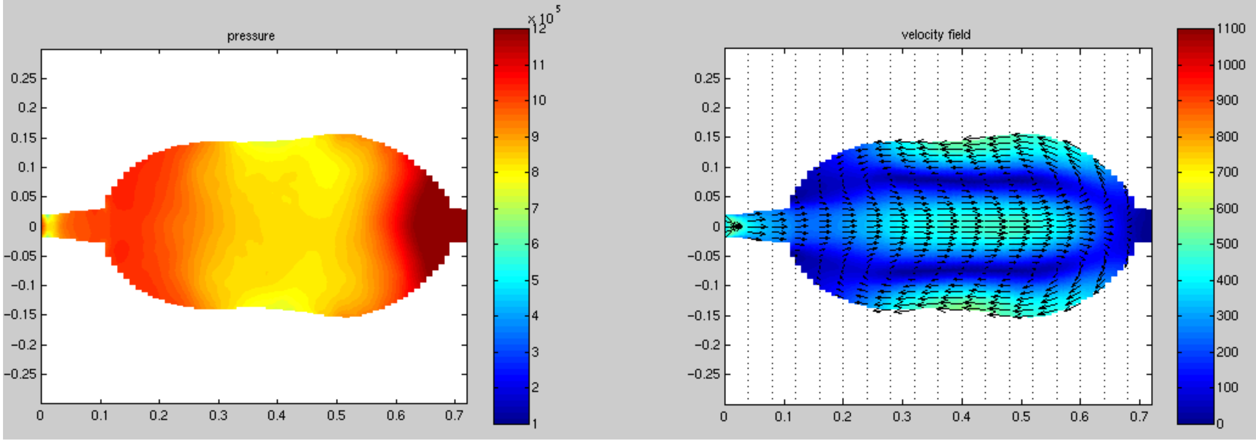


Fig. A.30 Velocity and pressure at time $t = 0.006s$

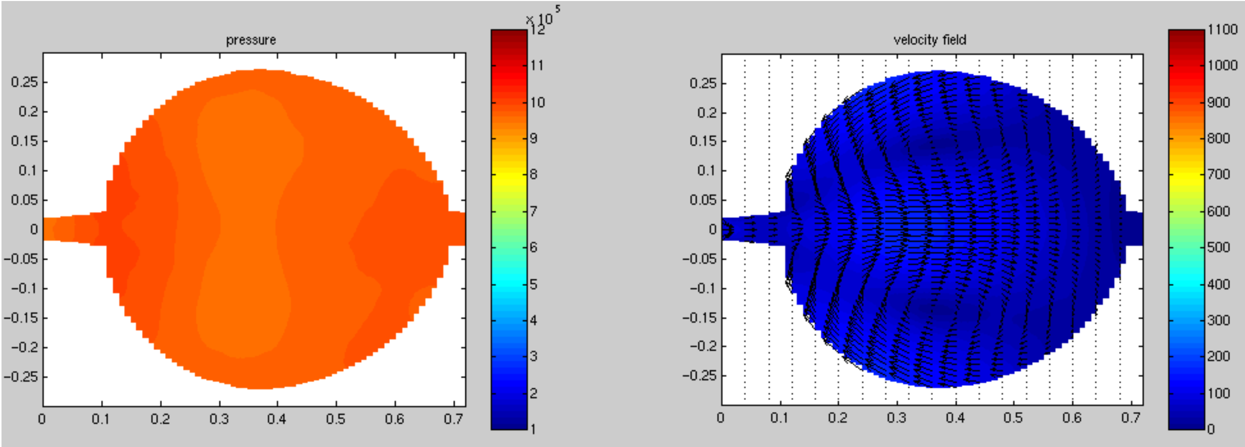


Fig. A.31 Velocity and pressure at time $t = 0.025s$

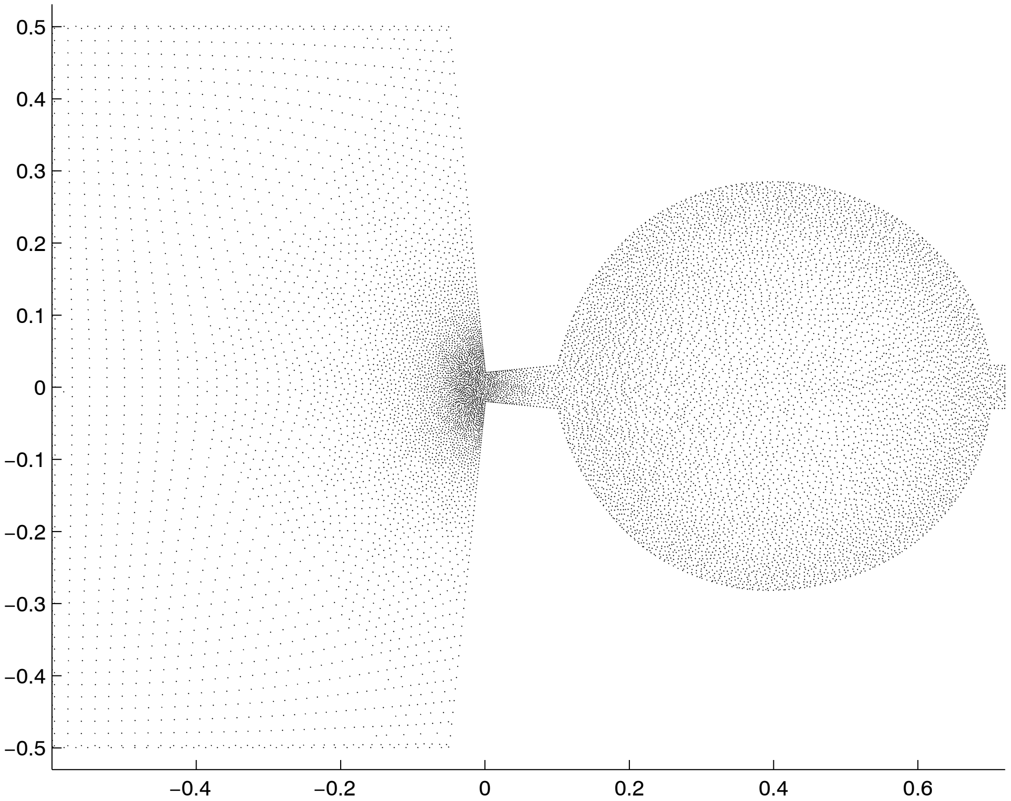


Fig. A.32 Particle distribution at time $t = 0.025s$

Jörg Kuhnert

Curriculum Vitae

10. Juli 1969	geboren in Karl-Marx-Stadt (jetzt Chemnitz)
1976 - 1978	Polytechnische Zehnklassige Allgemeinbildende Oberschule (POS) 'Oleg Koschewoi' K.-Marx-St.
1978 - 1982	POS 'Walentina Tereschkowa' K.-Marx-St., Schule mit erweitertem Russischunterricht
1982 - 1984	Kinder- und Jugendsportschule (KJS) 'Emil Wallner' K.-Marx-St.
1984 - 1986	POS 'Walentina Tereschkowa' K.-Marx-St., Schule mit erweitertem Russischunterricht
Juli 1986	Abschluß 10. Klasse
1986 - 1988	Arbeiter-und-Bauern-Fakultät 'Walter Ulbricht' Halle/Saale
Juli 1988	Abitur
1990 - 1992	Vordiplomstudium Maschinenbau RWTH Aachen
Sept. 1992	Vordiplom Maschinenbau
1992 - 1996	Hauptstudium Maschinenbau mit Vertiefungsrichtung Energietechnik RWTH Aachen
1994	einsemestriger Aufenthalt an der Firma 'KADE-TECH Recherche' Lyon, Frankreich
1994 - 1995	Masterstudiengang Numerische Mathematik an der Oregon State University, Corvallis, USA
Dez. 1995	Master of Science
seit Sept. 1996	Dissertation zum Thema 'Smoothed Particle Hydrodynamics' an der Universität Kaiserslautern im Rahmen des 'Graduiertenkollegs Technomathematik'
Juli 1997	Diplom Maschinenbau RWTH Aachen
seit Juli 1997	wissenschaftlicher Mitarbeiter am Institut für Techno- und Wirtschaftsmathematik Kaiserslautern
1998	zweimonatiger Aufenthalt am Indian Institute of Science, AR & DB Centre of Excellence for Aerospace CFD, Bangalore, India