

ON THE USE OF MESHLESS METHODS IN ACOUSTIC
SIMULATIONS

by

Philipp Hahn

A thesis submitted in partial fulfillment of the requirements for the
degree of

Master of Science
(Mechanical Engineering)

at the

UNIVERSITY OF WISCONSIN – MADISON

2009

Table of Contents

Abstract	i
Acknowledgments	iii
List of Figures	iv
1 Introduction.....	1
2 Technical Background.....	5
2.1 The Linear Acoustic Wave Equation	6
2.1.1 Derivation	7
2.1.2 Limits	9
2.1.3 Solution Methods	10
2.2 The Navier-Stokes Equation	11
2.2.1 Limits	13
2.2.2 Solution Methods	13
3 Acoustic Lumped Mass Model for Ideal Gas	15
3.1 Equation of Motion for the One-Dimensional Lumped Mass Model.....	16
3.2 Model Parameters & Relations to the Acoustic Wave Equation	18
3.3 Implementation	20
3.4 Numerical experiments	23
3.4.1 Open ended tube with impulse excitation	24
3.4.2 Triangular initial pressure contour	25
3.4.3 Nonlinear examples	27
4 Smoothed Particle Hydrodynamics.....	35
4.1 The Basic SPH Formalism.....	37
4.2 SPH Formulation for Fluid Dynamics and Acoustic Simulations	42
4.2.1 Conservation of mass	43
4.2.2 Conservation of momentum.....	43
4.2.3 Conservation of energy	46
4.2.4 Equation of State.....	48
4.2.4.1 Ideal Gas	48
4.2.4.2 Incompressible Liquids	49
4.3 Particle motion	50
4.4 Time integration.....	51
4.5 The Smoothing Kernel	52
4.6 Variable Smoothing Length	53
4.7 Boundary Formulations	55
4.7.1 Dynamic Boundary Particles	55
4.7.2 Repulsive Forces	56
4.7.3 Mirroring Particles	57
4.7.4 First Principal Boundary Conditions	59
4.7.5 Simplified First Principle Boundary Conditions.....	60
4.8 Fluid-Structure Interaction.....	66
4.9 SPH for Acoustic Simulations	69
5 Implementation of SPH	73

5.1	One and Two-Dimensional Implementation.....	76
5.1.1	2-D Hydrodynamic Test Simulation.....	78
5.2	Three-Dimensional Implementation	80
5.2.1	GPU-Programming	82
5.2.2	Program Structure	86
5.2.3	3-D Hydrodynamic Test Simulations	88
6	Numerical Experiments.....	93
6.1	Two Dimensional Sound Propagation	94
6.2	The Smoothing Length and its Impact on Wave Speed.....	100
6.3	Sound excitation and propagation in a Tube.....	108
6.4	Work Precision Diagram.....	118
7	Conclusion and Future Work	125
	References.....	128

Abstract

Philipp Hahn

Under the Supervision of Dr. Dan Negrut

at the University of Wisconsin – Madison

Numerical simulation of physical phenomena is gaining popularity in all fields of engineering as computing capabilities grow rapidly. Acoustical engineering is a domain which is increasingly gaining momentum because cheap and fast numerical simulations are replacing costly experiments. Despite tremendous improvements on existing simulation methods, various acoustic problems are still particularly challenging to model.

Currently, the most common techniques for acoustic simulations draw on mesh-based methods such as the Boundary Element Method (BEM), Finite Differences Method (FD) and Finite Element Method (FEM), which are typically used to solve the linear acoustic wave equation. In theory, this equation is limited to linear elastic propagation media. However, almost all real propagation media display nonlinear characteristics as soon as pressure amplitudes or gradients are high. Shock or ultrasound wave propagation through gas and sound propagation through highly nonlinear materials are examples that cannot be modeled by the linear acoustic wave equation. Furthermore, many difficulties related to a mesh-based problem formulation limit applications of current methods. Complicated domain topologies

and moving or deformable boundaries are particularly hard to describe accurately in this class of methods.

Therefore, this work investigates the potential and limitations of two Lagrangian particle methods for acoustic simulations. Specifically, these alternative formulations are a lumped mass model and a meshless Lagrangian technique, called Smoothed Particle Hydrodynamics (SPH).

The lumped mass model is successfully used to expand linear wave propagation with nonlinear characteristics. Due to its applicability to only one-dimensional problems, a second and more promising method, SPH, is studied in detail. In contrast to currently used methods, SPH is employed to solve the compressible Navier-Stokes Equations to resolve sound propagation and fluid flow simultaneously. Moreover, effects due to nonlinear propagation media are modeled accurately since no linearizations are made. By virtue of its meshless nature, SPH can handle complicated and time-variant domain topologies without the typical problems of grid-based methods. Furthermore, SPH particles carry material properties, making it easy to simulate the propagation of sound through multiphase fluid flows or gases with different wave speeds. It is also well suited to simulating sound propagation through gases with a temperature gradient resulting in a gradient in the speed of sound. A very attractive feature of SPH is the intuitive handling of fluid-structure interactions. This can be used to simulate sound excitation due to moving structural elements as well as sound-induced structure motion.

The investigation of SPH for simulating sound propagation is carried out in order to assess its potential in relation to the limitations associated with existing simulation methods. Computational experiments carried out for the verification of the new approach indicate that

SPH models sound propagation accurately. Furthermore, sound excitation due to moving boundaries shows good agreement with analytic solutions. The motion of a rigid body due to fluid excitation is also simulated, demonstrating the potential applications of the method. The simulations presented draw on a fast, massively parallel GPU implementation of the SPH algorithm which is also discussed briefly.

Acknowledgments

First of all, I would like to acknowledge to my advisor, Assistant Professor Dan Negrut, who greatly supported me during my time in the Simulation-Based Engineering Laboratory (SBEL) and gave me the opportunity to study this fascinating subject with the freedom to choose research directions. Without his hints and suggestions this work would have been impossible to complete. I would like to thank the other members of the committee, Professor Michael R. Zinn and Professor Darryl G. Thelen, for their expertise. I am also very grateful for the assistance I received from my colleagues in the Simulation-Based Engineering Laboratory. Finally, I would also like to thank my roommates Martin Tupy and Bastian Niebel for their help and the inspiring discussions while having lunch.

List of Figures

Figure 1.1:	The steps necessary to set up and use a numerical simulation.	2
Figure 3.1:	Basic one-dimensional lumped mass model for ideal gas with masses and springs.	15
Figure 3.2:	One dimensional lumped mass model with coordinate system.	16
Figure 3.3:	Visualization of the one-dimensional lumped mass model for an example with nine masses.	22
Figure 3.4:	Amplitude plot of the pressure signal (top) and FFT-magnitude plot of the same signal (bottom).	24
Figure 3.5:	Pressure contour recorded at four different times (given at the top of each plot). The initial pressure distribution has a triangular shape.	26
Figure 3.6:	Pressure between the first two masses of the lumped mass model plotted over time.	27
Figure 3.7:	Pressure contours simulated with the linearized lumped mass model given. The four pressure graphs are given at different times for zero initial conditions and sinusoidal excitation from the right side.	28
Figure 3.8:	Pressure contours simulated with the nonlinear lumped mass model given. The four pressure graphs are given at different times for zero initial conditions and sinusoidal excitation form the right side.	29
Figure 3.9:	Temporal development of an initial sinusoidal wave into a sequence of solitary peaks calculated using the Korteweg-de Fries equation.	30
Figure 3.10:	Simulation of three soliton waves traveling through a 1 meter long tube with closed ends. The propagation speed of the solitons increases with amplitude. Due to this effect, higher solitons overtake the smaller ones.	33
Figure 4.1:	This figure illustrates the Kernel, W , and its support domain, Ω , which is bordered by S . The field function at every point (not necessarily at a particle) can be approximated by summing up the weighted function values at the positions of the particles in the support domain [26].	38
Figure 4.2:	This figure shows the kernel function, W , in two dimensions. The surface integral in Eq. (56) is equal to zero as long as the kernel function is compact or in other words equal to zero at the surface or boundary of the support domain.	40
Figure 4.3	Kernel function, W , close to the domain boundary. In case the support domain of the kernel overlaps the boundary of the problem domain the surface integral in Eq. (56) can be non-zero.	41
Figure 4.4:	Dynamic boundary particles placed outside the domain to remove boundary deficiency. The thickness of this boundary particle layer has to be twice the smoothing length. Unlike fluid	

particles, dynamic boundary particles do not move relative to the boundary.	56
Figure 4.5: Location and velocity of mirror particles. The support domain of particle i overlaps the domain boundary and potential particle deficiency is avoided through mirror particles.	57
Figure 4.6: The mirror particle boundary formulation does not completely remove particle deficiency if the boundary has sharp edges (left) or is curved (right).	58
Figure 4.7: Interaction of particles j and i over field variable values on the boundary. The pressure, p_b , at one point of the boundary depends on properties of particle j . Since the point on the boundary is within the support domain of particle i , the particles interact indirectly.	61
Figure 4.8: Surface integral over the two-dimensional SPH kernel, W , on the domain boundary.	64
Figure 4.9: Surface integral over the normalized two-dimensional cubic spline kernel and the approximation function.....	65
Figure 4.10: Surface integral over the normalized three-dimensional cubic spline kernel and the approximation function.....	66
Figure 5.1: Binning step of spatial subdivision. Since a cubic spline kernel is used, the bin side length has to be equal to $2h$ in order to obtain a fast algorithm along with capturing every possible particle interaction.	74
Figure 5.2: Basic SPH formulation. After the particles have been created their time evolution is simulated in the simulation loop while data can be saved for further analysis.	76
Figure 5.3: Hierarchic structure of the Matlab m-files of the SPH program.....	77
Figure 5.4: Pressure wave propagation and free surface flow due to gravity in the dam break experiment.....	80
Figure 5.5: Speed comparison between GPUs and CPUs in terms of FLOP rate [53]. While for the CPU the Flop rate is given for double precision arithmetic, the values for the GPU have been evaluated for only single precision computation. However, the slope of the GPU FLOP rate is significantly higher.....	82
Figure 5.6: Hardware structure of GPUs [53].	83
Figure 5.7: Memory structure of GPUs [53].	84
Figure 5.8: Structure of a CUDA program [53].	85
Figure 5.9: Time evolution of particle positions rendered in Povray showed at nine different time steps. The droplet falls into a basin of liquid and creates a characteristic impact crater.	89
Figure 5.10: Photo of a “water crown” formed after the impact of a water droplet (source: DHD Multimedia Gallery).....	90
Figure 5.11: The interaction between fluids of different viscosities modeled using about 70,000 SPH particles.....	91
Figure 5.12: Fluid-structure interaction for the example of a trough that due to hydrodynamic forces starts to rotate around a pivot axis. The	

fluid is modeled with 250, 000 SPH particles and the trough consists of 6760 boundary particles	92
Figure 6.1: Surface plots of the pressure field obtained by FDTD (left column) and SPH (right column) for time steps 0.053 ms, 0.100 ms and 0.150 ms.....	98
Figure 6.2: Surface plots of the pressure field obtained by FDTD (left column) and SPH (right column) for time steps 0.200 ms, 0.250 ms and 0.400 ms.....	99
Figure 6.3: Sound pressure plot for smoothing length equal 1 mm. Very good agreement with the analytical solution.	104
Figure 6.4: Sound pressure plot for smoothing length equal 1.7 mm. Wave speed is slightly higher compared to the analytical solution.	104
Figure 6.5: Sound pressure plot for smoothing length equal 1.5 mm. The wave speed is lower than the theoretical value.	104
Figure 6.6: Sound pressure plot for smoothing length equal 2 mm. The wave speed agrees with the analytical solution.	104
Figure 6.7: Sound pressure plot for smoothing length equal 3 mm. The sound speed agrees with the analytical solution but the quality of the numerical solution begins to suffer.....	105
Figure 6.8: Sound pressure plot for smoothing length equal 5 mm. The sound speed agrees with the analytical solution but the quality of the numerical solution is poor.....	105
Figure 6.9: Wave speeds of all 41 SPH simulations plotted over the smoothing length. Convergence of the SPH wave speed towards the theoretical value can be observed with growing smoothing length.....	105
Figure 6.10: Adimensionalized version of Figure 6.9. The smoothing length divided by the particle spacing is plotted on the x-axis and the relative error in the wave speed is plotted on the y-axis.....	106
Figure 6.11: FDTD solution for the same experiment. The wave speed agrees with the analytical solution.	107
Figure 6.12: Geometry and coordinate axis of the tube. It has a quadratic cross section and is filled with ideal gas.	108
Figure 6.13: Pressure plotted for five subsequent times (given at the top of each plot) over the position on a horizontal level at $y = 0$ m. The wave, excited by piston motion, travels from the left end of the tube to the right and gets reflected. Dynamic boundary particles are used.	111
Figure 6.14: This graphic shows a representation of the tube cross section. Left: Particle deficiency at sharp corners when using the mirror particle approach; Right: No particle deficiency due to the use of dynamic boundary particles.	112
Figure 6.15: Pressure loss at sharp corners when using mirror particles. The plots show the pressure (represented as color) distribution over the cross section of the tube for four subsequent time steps.....	113

Figure 6.16:	Patches of dynamic boundary particles are placed in the corners to prevent particle deficiency.	114
Figure 6.17:	Pressure plotted for five subsequent times (given at the top of each plot) over the position on a horizontal level at $y = 0 \text{ m}$. The wave, excited by piston motion, travels from the left end of the tube to the right and gets reflected. A combination of mirror and dynamic boundary particles is used to describe the rigid walls.	115
Figure 6.18:	Pressure loss at the surface of a tube with circular cross section. The mirror particles produce disturbances and do not prevent boundary penetration.	117
Figure 6.19:	Pressure over position after $8e^{-4}$ seconds in a horizontal slice at level $z = 0 \text{ m}$. The resolution is given in particles per meter on top of each plot.	119
Figure 6.20:	Reference solution obtained using FDTD. The resolution is here 176 cells per meter.	120
Figure 6.21:	Arithmetic mean of the relative error, measured at 1,000 different positions in the domain.	121
Figure 6.22:	Computing time for one simulation step, plotted over the number of SPH particles in the simulation. The times are recorded for the simulations above on a NVIDIA GeForce 8800 GTX graphic card.	122
Figure 6.23:	Total simulation time on a NVIDIA GeForce 8800 GTX graphic card, recorded for the simulations above.	123
Figure 6.24:	Work-precision diagram for the acoustic SPH simulations above. The achieved precision is plotted over the total simulation time on a NVIDIA GeForce 8800 GTX graphic card.	123

Chapter 1

Introduction

Numerical simulations of dynamic systems are a valuable tool to predict the way systems change in time and provide deeper insights into the dynamics of processes. These simulations are becoming increasingly important for the study of complex problems in engineering and many other fields like physics and chemistry. In engineering, the development of new products or new technologies often leads to questions that have to be analyzed in detail. Many of these questions can be answered by both experiments and simulations. However, simulations are typically more cost effective. The rapid increase of computational power during the past years has led to much more affordable simulation hardware with high computing capabilities. This development will lead to a wider use of numerical simulation in all fields of engineering and natural science. Additionally, in some cases it is simply not possible to conduct physical experiments since they are not feasible using current technology, would result in safety hazards or would require unreasonable efforts and costs.

In general, numerical simulations follow a basic procedure as illustrated in Figure 1.1.

NUMERICAL SIMULATION

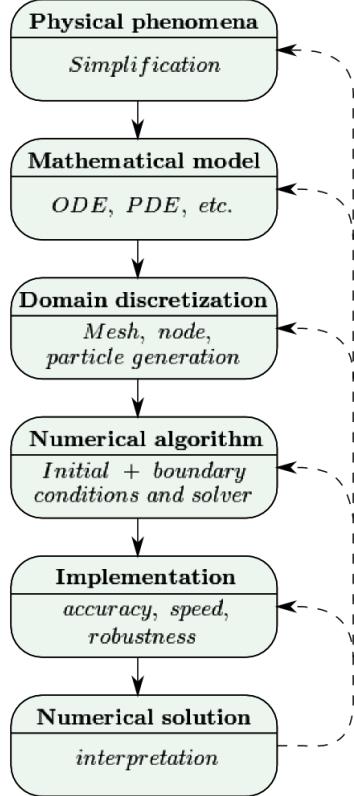


Figure 1.1: The steps necessary to set up and use a numerical simulation.

The starting point is a physical phenomenon that needs to be simulated in the process of developing a new product or new technology. In the second step this physical phenomenon has to be described by a simplified mathematical model which captures all important physical aspects. Differential equations are widely used for this purpose. For very simple problems the mathematical model can be solved directly. However, in most cases, a domain discretization has to be utilized in combination with a numerical solution algorithm to carry out a simulation. Subsequently, the algorithm needs to be implemented in a computer language in order to leverage the computing power of modern numerical computers. The last step in the sequence is an interpretation of the numerical results. All steps combined

result in a numerical simulation. However, the whole process is often not straightforward and iterative changes have to be made in the different stages to obtain good simulation results. Once a new simulation method is implemented it is therefore essential to validate results using experimental data, analytic solutions or other well established simulation methods. In this way information about accuracy, efficiency and robustness can be obtained.

The physical phenomenon of interest in this work is sound wave propagation through elastic media like gases or liquids. Dependent on the medium, different types of wave propagation can be observed [1].

The microstructure of solids consists of atoms or molecules that are organized in a closely packed structure that doesn't change significantly with time as long as the deformation due to pressure wave propagation is small. Particles cannot move freely but they can oscillate around an equilibrium position. For example, in solid metals the atoms are organized in crystal structures. The organization into closely packed structures is caused by intermolecular or inter-atomic forces which are also responsible for the elastic properties of the material. If one or more atoms of the structure are perturbed from their equilibrium positions the inter-atomic forces will cause a displacement of the neighboring atoms which again results in displacements of other neighboring atoms and so on. This mechanism results in waves which travel through the material. From the atomistic perspective there exists only attraction or repulsion forces between two atoms, but since the particles are closely packed, solid materials also support shear forces. This is the reason why different macroscopic wave propagation mechanisms are distinguished for solid media. These are the two basic longitudinal and transversal, or shear modes, along with various other forms. Sound

propagation through solids is closely related to the theory of mechanical vibration and won't be discussed further in this thesis.

The microscopic structure of liquids or gases is significantly different than that of solids. Since the kinetic energy of the atoms is high enough to overcome inter-atomic attraction forces, atoms are not organized in static structures. Due to the fact that the particles in low-viscosity liquids can easily move relative to each other, they cannot sustain high shearing forces and therefore support shear and bending wave propagation only to an insignificant extent. Therefore, only longitudinal sound wave propagation has to be considered as long as the fluid is not highly visco-elastic.

Particles in gases have even more kinetic energy than those in fluids and the distance between the particles is very high in comparison to the particle size. As a result, interatomic forces become negligible over those long distances and the motion of the particles is mostly governed by collisions. Transversal sound wave propagation can again be neglected due to the fact that gases typically have extremely low viscosity.

This work mainly addresses sound propagation through gases for which the technical background is given Chapter 2 together with commonly used simulation methods. Chapter 3 describes the development of the lumped mass model for nonlinear sound propagation. A detailed discussion about SPH follows in Chapter 4. Two different implementations of the SPH algorithm are described in Chapter 5. Chapter 6 contains numerical experiments conducted in order to validate results, obtained using SPH, against other more established acoustic simulation methods or analytical results. Conclusions along with future research directions are provided in Chapter 7.

Chapter 2

Technical Background

There exist different methods that describe gases on a microscopic level. Most commonly used are Cellular Automata, Direct Simulation Monte Carlo (DSMC), the Lattice Boltzmann method, Brownian Motion and the direct calculation of intermolecular collision forces in Molecular Dynamics simulations (MD).

However for most engineering application only the macroscopic quantities such as bulk velocity, pressure or temperature are of interest. It is interesting to note that they are simply our macroscopic perceptions of the potential and kinetic energy stored in the microscopic particles. In order to capture the dynamics and the relations between the macroscopic quantities, mathematical models have been developed. Three fundamental physical conservation laws for mass, momentum and energy are the basis of these models.

$$1) \text{ Conservation of mass:} \quad \frac{d\rho}{dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta} \quad (1)$$

$$2) \text{ Conservation of momentum:} \quad \frac{dv^\alpha}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} + f^\alpha \quad (2)$$

$$3) \text{ Conservation of energy:} \quad \frac{du}{dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{\partial v^\alpha}{\partial x^\beta} \quad (3)$$

where α denotes the coordinate direction and β stands for repeated indices. In these equations, ρ stands for density, v for velocity, σ for stress, u for internal energy and f^α for body forces. The conservation laws are used in combination with constitutive relations to

derive the Navier-Stokes equation for fluid and gas dynamics and also the wave equation for acoustics. For Newtonian fluids, the total stress tensor, $\sigma^{\alpha\beta}$, consists of the isotropic pressure, p , and the viscous stress $\sigma_{visc}^{\alpha\beta}$ as shown in Eq. (4).

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \sigma_{visc}^{\alpha\beta} \quad (4)$$

The compressible Navier-Stokes equation comprises constitutive equations for compressibility and viscosity as well as conservation of mass, momentum and sometimes energy, whereas the wave equation is based only on a constitutive equation for compressibility and the conservation laws of mass and momentum (see section 2.1.1). This shows the close connection between acoustics and fluid mechanics. In fact, the compressible Navier-Stokes equation can be used to simulate acoustic wave propagation through elastic media. However, other than the wave equation, the mathematical model of the Navier-Stokes equations captures all important physical aspects of the whole flow process. Consequently, for problems in which the flow process affects the sound propagation, only the Navier-Stokes equation is feasible to set up a simulation. Specific problems where the solution of the wave equation is not sufficient will be discussed later in this chapter.

2.1 *The Linear Acoustic Wave Equation*

Most problems in fields such as psychoacoustics, engineering acoustics and architectural acoustics can be addressed solving the linear wave equation with an appropriate numerical algorithm. The wave equation is a hyperbolic partial differential equation was first derived in one-dimensional form for vibrating strings and later enhanced for multidimensional problems. Renowned scientists like d'Alembert, Euler, Bernoulli, and Lagrange made important contributions studying the wave equation. The wave equation in its general form,

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u , \quad (5)$$

with field variable, u , and speed of sound, c , has numerous application in different fields of dynamics. The acoustic wave equation describes the evolution of acoustic pressure, \tilde{p} , or velocity, \tilde{v} , as a function of time and position,

$$\frac{\partial^2 \tilde{p}}{\partial t^2} = c^2 \nabla^2 \tilde{p} ; \quad \frac{\partial^2 \tilde{v}}{\partial t^2} = c^2 \nabla^2 \tilde{v} . \quad (6)$$

2.1.1 Derivation

In order to assess its limits, the one-dimensional linear wave equation will be derived from the conservation of mass and momentum as well as the ideal gas law.

Assumed is a fast adiabatic compression of ideal gas without heat transfer. The relation between density, ρ , and pressure, p , before and after the process can then be described by,

$$p_2 = p_1 \cdot \left(\frac{\rho_2}{\rho_1} \right)^\kappa . \quad (7)$$

Linearization around p_1 leads to Eq. (8),

$$p_2 - p_1 = \left(\frac{p_1 \kappa}{\rho_1} \right) \cdot (\rho_2 - \rho_1) . \quad (8)$$

If p_1 and ρ_1 represent mean conditions, an equation can be found that relates small acoustic pressure and density variations \tilde{p} and $\tilde{\rho}$, whereas p and ρ are the total pressure and density,

$$\underbrace{p - p_{mean}}_{\tilde{p}} = \underbrace{\frac{p_{mean} \kappa}{\rho_{mean}}}_{\frac{\partial \tilde{p}}{\partial \tilde{\rho}}} \cdot \underbrace{(\rho - \rho_{mean})}_{\tilde{\rho}} . \quad (9)$$

The one-dimensional continuity equation,

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho v) = 0 , \quad (10)$$

can be used in combination with the expression for the total density, $\rho = \rho_{mean} + \tilde{\rho}$, to relate the acoustic density variation to the velocity gradient,

$$\frac{\partial}{\partial t} (\rho_{mean} + \tilde{\rho}) + \frac{\partial}{\partial x} (\rho_{mean} v + \tilde{\rho} v) = 0. \quad (11)$$

Assuming that ρ_{mean} does not change over time and space, $\tilde{\rho}$ is small and the gradient of $\tilde{\rho}$ is small, we can simplify Eq. (11) to the following form,

$$\frac{1}{\rho_{mean}} \frac{\partial \tilde{\rho}}{\partial t} + \frac{\partial v}{\partial x} = 0. \quad (12)$$

Substituting the expressions for total density and pressure in the conservation law of momentum,

$$\rho \frac{dv}{dt} + \frac{\partial p}{\partial x} = 0, \quad (13)$$

leads to Eq. (14), which again can be simplified.

$$\underbrace{(\rho_{mean} + \tilde{\rho})}_{\rho} \left(\frac{\partial}{\partial t} + v \frac{\partial}{\partial x} \right) v + \frac{\partial}{\partial x} \underbrace{(p_{mean} + \tilde{p})}_{p} = 0 \quad (14)$$

Assuming small density and pressure variations, $\tilde{\rho}$ and \tilde{p} , small velocity gradients, $\frac{\partial v}{\partial x}$, and no changes of the mean pressure, p_{mean} , over space, Eq. (14) takes the following form,

$$\rho_{mean} \frac{\partial v}{\partial t} + \frac{\partial \tilde{p}}{\partial x} = 0. \quad (15)$$

In the following step Eqs. (9), (12) and (15) are combined to obtain the linear wave equation.

By taking the time derivative of Eq.(12),

$$\frac{1}{\rho_{mean}} \frac{\partial^2 \tilde{\rho}}{\partial t^2} + \frac{\partial^2 v}{\partial x \partial t} = 0, \quad (16)$$

and the spatial derivative of Eq. (15),

$$\rho_{mean} \frac{\partial^2 v}{\partial t \partial x} + \frac{\partial^2 \tilde{p}}{\partial x^2} = 0, \quad (17)$$

we can combine both equations and get a relation between acoustic pressure and density,

$$\frac{\partial^2 \tilde{\rho}}{\partial t^2} - \frac{\partial^2 \tilde{p}}{\partial x^2} = 0, \quad (18)$$

substituting the expression for acoustic density, $\tilde{\rho}$, we end up with the linear acoustic wave equation:

$$\underbrace{\frac{\rho_{mean}}{p_{mean} \cdot \kappa}}_{\frac{1}{c^2}} \cdot \frac{\partial^2 \tilde{p}}{\partial t^2} - \frac{\partial^2 \tilde{p}}{\partial x^2} = 0 \quad (19)$$

$$\Rightarrow c = \sqrt{\frac{p_{mean} \kappa}{\rho_{mean}}}.$$

The above equation can be combined similarly to an identical wave equation that describes the velocity, \tilde{v} , with respect to space and time.

$$\underbrace{\frac{\rho_{mean}}{p_{mean} \cdot \kappa}}_{\frac{1}{c^2}} \cdot \frac{\partial^2 \tilde{v}}{\partial t^2} - \frac{\partial^2 \tilde{v}}{\partial x^2} = 0 \quad (20)$$

2.1.2 Limits

As seen in the derivation, the linear acoustic wave equation is based on numerous assumptions. It is only valid for moderate sound pressure levels ($L_p \lesssim 120 \text{ dB}$ or $\tilde{p} \lesssim 20 \text{ Pa}$), no high pressure or density and no gradients in the mean pressure or mean density. Furthermore, this mathematical model does not include effects of friction, viscosity or anisotropic propagation media. Other effects, caused by the presence of an underlying bulk flow can't be simulated either.

Though most acoustic problems can be addressed by solving the linear wave equation, there are clearly documented difficulties with certain classes of applications. One of the challenging aspects is the modeling of moving boundaries [2]. The movement does not only change the way sound waves are reflected in a cavity, but it also generates sound and changes the density of the media in the cavity [3]. These density changes lead to different wave speeds or, in case of density gradients, to gradients in the wave speed. These effects become crucial as the deflection or the velocity of the boundary gets larger.

2.1.3 Solution Methods

Several classes of numerical techniques are available to solve the linear wave equation. One branch of methods solves another differential equation, called the Helmholtz equation (Eq. (21)), which is closely related to the wave equation.

$$\frac{\nabla^2 \tilde{p}}{\tilde{p}} = -k^2 \quad (21)$$

It can be obtained from the wave equation by applying the technique of separation of variables and represents the time invariant form of the original equation. Frequency-domain techniques such as FEM [4] and BEM [5] can solve the Helmholtz equation for the specific problem in a one-frequency-at-a-time fashion. This means that transfer functions can be obtained by solving the Helmholtz equation for a collection of frequencies. The spatial discretization which is used to transform the underlying differential equation into a set of linear algebraic equations has to be refined with higher frequencies. Thus the number of equations that have to be solved grows quickly with frequency, restricting FEM and BEM to small spatial domains or low frequencies [6, 7].

Time-domain techniques such as Finite-Differences Time-Domain (FDTD) [8], the digital waveguide mesh method (DWM) [9] and also different FEM and BEM formulations of the wave equation lead to a solution that comprises both the discretized time evolution and the spatial distribution of the sound pressure. For many applications these time-domain methods have computational advantages. Important room acoustic quantities such as the reverberation time or the impulse response can be obtained easily. Unfortunately complex shapes can be difficult to accommodate with the FDTD method [2].

Another branch of techniques such as ray-, beam-, particle- and cone tracing is summarized under the term Geometrical-Acoustics [10]. These methods are extensively used in room acoustics since they are computationally less expensive.

Though most problems can be solved sufficiently with these conventional methods, there are still cases in which solutions are difficult to find. Sound propagation can for example be affected by the varying sound speeds of different propagation media. Examples include the propagation of sound through the mixing flow of two different fluids or sound propagation through a medium of temporally or spatially changing temperature. Conventional methods have problems with the discretization of those problems and need to be coupled with separate Computational Fluid Dynamic (CFD) simulations.

2.2 The Navier-Stokes Equation

Named after the Scientists Claude-Louis Navier and George Gabriel Stokes, the Navier-Stokes equation describes the conservation of linear momentum for fluid flow [11]. Several versions of this equation for different liquids and gases are widely used in CFD simulations. The two most common versions for incompressible and compressible fluids however assume

Newtonian viscosity. It was observed that even turbulent flows can be simulated in good agreement with experimental results.

Eq. (22) shows the compressible Navier-Stokes equation that is based on the conservation laws for momentum, the Newtonian constitutive equation and an equation that describes compressibility.

$$\left(\frac{\partial(\rho\mathbf{v})}{\partial t} + \mathbf{v} \cdot \nabla(\rho\mathbf{v}) \right) = -\nabla p + \mu\nabla^2\mathbf{v} + \left(\frac{1}{3}\mu + \mu^v \right) \nabla(\nabla \cdot \mathbf{v}) + \mathbf{f} \quad (22)$$

The viscosity is given by μ , μ^v is the bulk viscosity that is responsible for sound wave attenuation as waves propagate through the fluid and \mathbf{f} accounts for body forces like gravity. Along with Eq. (22), the conservation of mass and, in some cases where fluid compression causes an increase of temperature, the conservation of energy are necessary to set up a simulations. The constitutive equation, the equations of state and the conservation laws are often simulated separately for each time step since solution algorithms usually do not require a combined formulation.

Aero-acoustics [12], the sound generation due to fluid motion, is an area of acoustics where it is not sufficient to solve the linear wave equation. One example is the sound generation due to the fluid dynamic excitation of light boundary structures. Another example is the flow over a cavity (blowing over a bottleneck) that can cause resonance effects and sound generation as well as turbulent fluid motion that generates sound itself. For this branch of acoustic problems the entire flow process has to be solved using the compressible Navier-Stokes equation rather than the wave equation.

2.2.1 Limits

The Navier-Stokes equations describe fluid motion in its full nonlinearity and the equation of state and the constitutive equation for viscosity are not required to be linear. Finding their numerical solution is demanding in most cases. However, the conservation equations and the constitutive equations can also be solved separately for every time step, which makes this formulation very flexible.

The Navier–Stokes equations are based on the assumption that the fluid is a continuum. This assumption holds for low Knudsen numbers which describe the ratio of mean free path to a length scale. High Knudsen numbers can occur for shockwaves, high frequency sound or rarefied gases. Depending on the problem, methods of statistical mechanics that describe the medium on a microscopic level have to be applied [13].

Since they don't account for heat conduction, the Navier-Stokes equations are also limited to applications where heat conduction doesn't play an important role. The time scales in acoustics are usually so small that heat conduction doesn't corrupt simulation results.

2.2.2 Solution Methods

Numerous discretization methods can be applied in combination with a solver in order to obtain solutions to the Navier-Stokes equation for a particular problem. The most commonly used discretization methods are Finite Volumes, Finite Elements, Boundary Elements and Finite Differences. In contrast to these mesh-based methods there exist meshless alternatives of which the radial basis method, Smoothed Particle Hydrodynamics (SPH) and different spectral methods are most commonly used.

The simulation of turbulent flows requires particular attention. It is possible to simulate turbulence with common discretization methods, however, all relevant length scales have to be resolved. This makes these so called Direct Numerical Simulations (DNS) [14] computationally extremely demanding for most problems. Reynolds-averaged Navier-Stokes equations (RANS) and Large-eddy simulations (LES) are the methods commonly used instead.

In aero-acoustics both the sound generation due to fluid flow and the sound propagation are of interest. A solution of the compressible Navier-Stokes equations for the specific problem can give the answer to both questions. For many aero-acoustic problems, analytical solutions and approximations for the sound generation have been found but their application is limited to the particular aerodynamic problem [15]. Another approach to solve these problems is DNS, which can be considered the most general formulation. The biggest shortcoming of this method is the extremely large problem formulation. Since length scales of acoustic variables are typically much higher than those of the bulk flow, discretizations need to be very fine.

The Hybrid Approach overcomes this problem by splitting the problem into a fluid dynamic and an acoustic domain. Two different solvers are used for the flow field and the acoustic field. The flow field can then be used to calculate acoustical excitations [16]. Unlike DNS, hybrid methods usually cannot account for changes in the speed of sound due to the bulk flow speed since they use a theoretical solution of the wave equation, making this approach less general.

Chapter 3

Acoustic Lumped Mass Model for Ideal Gas

Given the drawbacks of the linear acoustics wave equation, as seen in section 2.1.1, a lumped mass model for ideal gas has been developed that utilizes the ideal gas law, the conservation of momentum and the conservation of mass in their original forms without any linearization. The basic one-dimensional model structure is shown in Figure 3.1.

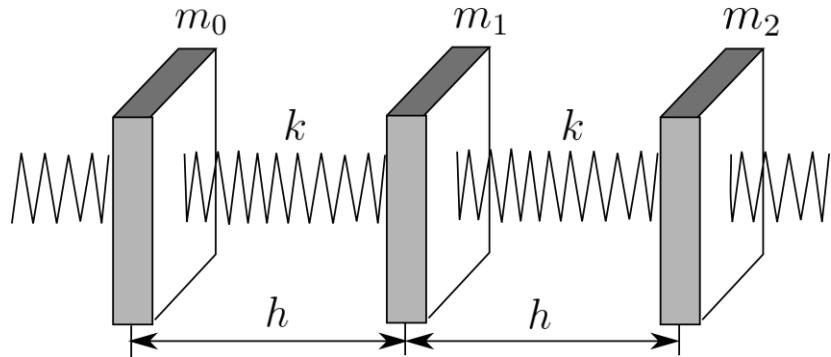


Figure 3.1: Basic one-dimensional lumped mass model for ideal gas with masses and springs.

Masses represent the inertia of a certain volume of air particles and springs model pressure forces on the same volume. It will be shown in section 3.1 that an infinitely fine discretization along with a linearization of the lumped mass model will lead to the following equation of motion,

$$\frac{\partial^2 d(x, t)}{\partial t^2} = \frac{KL^2}{M} \frac{\partial^2 d(x, t)}{\partial x^2}. \quad (23)$$

The similarity to the linear acoustic wave equation,

$$\frac{\partial^2 \tilde{v}}{\partial t^2} = \frac{p_{mean} \cdot \kappa}{\rho_{mean}} \frac{\partial^2 \tilde{v}}{\partial x^2}, \quad (24)$$

is obvious and for a properly chosen set of model parameters equivalents to velocity, \tilde{v} , and acoustic pressure, \tilde{p} , can be found.

3.1 Equation of Motion for the One-Dimensional Lumped Mass Model

Model

The one dimensional lumped mass model can be used to describe planar sound wave propagation through a tube. As mentioned above spring forces are used to replace pressure forces, whereas the springs have a nonlinear characteristic in order to represent the nonlinear relation between density, ρ , and pressure, p , for adiabatic processes. The model structure can be seen in Figure 3.2.

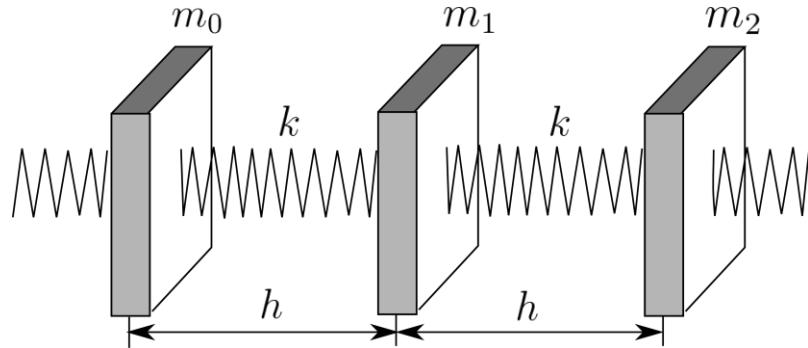


Figure 3.2: One dimensional lumped mass model with coordinate system.

In equilibrium position the masses are evenly distributed over x with a spacing of h . Displacements of the masses are expressed by $d_i(t)$ in a Lagrangian way.

Newton's Second Law for the middle mass m_1 leads to the following equation of motion,

$$\frac{\partial^2 d_1(t)}{\partial t^2} m_1 = \underbrace{F_l(d_0(t) - d_1(t), h) - F_r(d_1(t) - d_2(t), h)}_{\text{Sum of forces on mass point } m_0}, \quad (25)$$

where F_l and F_r are the spring forces to the left and to the right of mass m_1 , which depend nonlinearly on the respective spring elongation. In the following step Eq. (25) will be linearized in order to compare it to the linear acoustic wave equation.

The linearized version of Eq. (25) can be found by assuming a tangential spring stiffness of \tilde{k} at even equilibrium spacing h .

$$\frac{\partial^2 d_1(t)}{\partial t^2} m_1 = \tilde{k} \cdot (d_0(t) - d_1(t) + d_2(x, t) - d_1(t)) \quad (26)$$

Using the notation,

- Overall length: $L = N \cdot h$
- Total mass: $M = N \cdot m$
- Total stiffness of all springs in series: $K = \frac{\tilde{k}}{N}$

it follows that,

$$\frac{\partial^2 d_1(t)}{\partial t^2} = \frac{KL^2}{M} \cdot \frac{d_0(t) - 2d_1(t) + d_2(t)}{h^2} \quad \left| \text{discretized equation.} \right. \quad (27)$$

Therefore, an Eulerian formulation of the continuous limit $N \rightarrow \infty$ ($h \rightarrow 0$) can be obtained replacing the last term by the second spatial derivative.

$$\frac{\partial^2 d(x, t)}{\partial t^2} = \frac{KL^2}{M} \cdot \frac{\partial^2 d(x, t)}{\partial x^2} \quad \left| \text{continuous equation} \right. \quad (28)$$

Taking the time derivative of both sides leads to an expression for the velocity $u(x, t)$.

$$\frac{\partial^2 u(x, t)}{\partial t^2} = \frac{KL^2}{M} \cdot \frac{\partial^2 u(x, t)}{\partial x^2} \quad (29)$$

The discretization error of the lumped mass model is proportional to the difference between the difference quotient, $\frac{d_0(t) - 2d_1(t) + d_2(t)}{h^2}$, and the differential quotient, $\frac{\partial^2 d(x, t)}{\partial x^2}$. This means that

the model has second order accuracy and it will be most accurate for waves with low amplitude and frequency.

3.2 Model Parameters & Relations to the Acoustic Wave Equation

Comparing Eqs. (23) and (24), it is easy to see that model parameters for the lumped mass model have to be chosen in such a way that both wave speeds are equal. This means that the following equation has to be fulfilled:

$$\frac{M}{KL^2} = \frac{\rho_{mean}}{p_{mean}\kappa}. \quad (30)$$

The total mass M represents the mass of the medium in the total volume which can be calculated using the surface area, A , of the plates (Figure 3.2) and the total length, L , of the lumped mass model.

$$M = \rho_{mean}V = \rho_{mean} \cdot LA \quad (31)$$

The next parameter to be calculated is the total stiffness, K , of the spring array in the lumped mass model. It consists of small springs with the tangential stiffness, \tilde{k} , which are connected in series. Therefore a linearized expression for K can be found according to Eq. (32).

$$K = \frac{\tilde{k}}{N} \quad (32)$$

Since spring forces are used to replace pressure forces in the model, both must be equal to $F = p \cdot A$. Between two plates the adiabatic process given by Eq. (33) is assumed.

$$p = p_{mean} \left(\frac{\rho}{\rho_{mean}} \right)^\kappa \quad (33)$$

Density, ρ , can be expressed by the plate displacements left and right of the volume as shown in Eq. (34),

$$p = p_{mean} \left(\frac{h}{\underbrace{h + d_1 - d_0}_s} \right)^\kappa, \quad (34)$$

whereas s is the elongation of the small spring in between two plates. Therefore the nonlinear spring response of each spring in the model is given by,

$$F = A \cdot p_{mean} \left(\frac{h}{h + s} \right)^\kappa \quad \left| \begin{array}{l} \text{nonlinear spring response.} \end{array} \right. \quad (35)$$

Performing linearization around $s = 0$, Eq. (35) can be divided into a constant mean force, F_{mean} , and a displacement dependent term, \tilde{F} .

$$F = \underbrace{A \cdot p_{mean}}_{F_{mean}} - \underbrace{\frac{A \cdot p_{mean} \kappa s}{h}}_{\tilde{F}} \quad \left| \begin{array}{l} \text{linearized spring response} \end{array} \right. \quad (36)$$

The linearized spring stiffness is then given as $\tilde{k} = \frac{A \cdot p_{mean} \kappa}{h}$.

According to the equation for the total length, $L = h \cdot N$, it follows that the total stiffness should assume the value,

$$K = \frac{\tilde{k}}{N} = \frac{A \cdot p_{mean} \kappa}{hN} = \frac{A \cdot p_{mean} \kappa}{L}. \quad (37)$$

Substituting Eqs. (31) and (37) in Eq. (30) leads to,

$$\frac{M}{KL^2} = \frac{\rho_{mean} \cdot LA}{\frac{A \cdot p_{mean} \kappa}{L} \cdot L^2} = \frac{\rho_{mean}}{p_{mean} \kappa}. \quad (38)$$

This shows the equivalence between the continuous limit of the linearized lumped mass model and the linerized acoustic wave equation. However, if the lumped mass model is set

up using the nonlinear spring response, given by Eq. (35), simulation results will differ from solutions of the wave equation as soon as nonlinearities come into play.

Besides the spring response, the mass still needs to be determined. The masses m_i have to represent the mass of the medium, therefore they can be obtained as,

$$m_i = A \cdot \rho_{mean} h . \quad (39)$$

The area A of the plates can be freely chosen in 1D, however the two other parameters, ρ_{mean} and h , are specified by the propagation medium and the model discretization, respectively.

The most important field variable in acoustic simulations is typically pressure. Consequently, a relation between pressure and spring forces needs to be found. Such a relation can be found using the definition of pressure, $p = F/A$, in combination with Eq. (39). The equivalent pressure depends on spring force, F , discretization parameters, h and m , and the mean density of the medium, ρ_{mean} .

$$p = \frac{F}{A} = \frac{F \cdot \rho_{mean} h}{m} \quad (40)$$

3.3 Implementation

The equations described above are implemented in a Matlab program, which simulates the time evolution of this lumped mass model. Parallel to the springs, dampers have been included in order to create dissipative behavior. These dampers don't affect the sound propagation significantly since the damping ratio is very small. The spring- and damping forces applied to each mass are calculated with respect to the relative position and the velocity of the neighboring masses according to Eq. (35). Newton's second law is used as shown in Eq. (25) to calculate accelerations for each mass, which can then be integrated over

time. The basic Verlet integration algorithm was chosen for this task since it provides second order accuracy along with good stability [17]. Unlike the Euler forward integration algorithm, Verlet utilizes the central difference instead of a forward difference approximation. This central difference approximation is responsible for second order accuracy and it can be easily applied on the acceleration term, $a(t)$, as shown in Eq. (41).

$$a(t) = \frac{d^2\vec{x}(t)}{dt^2} \approx \frac{\vec{x}(t + \Delta t) - 2\vec{x}(t) + \vec{x}(t - \Delta t)}{\Delta t^2} \quad (41)$$

Multiplying Eq. (41) by Δt^2 and solving it for the position $x(t + \Delta t)$ leads to an expression for the position at the next time step,

$$x(t + \Delta t) \approx x(t) + v(t)\Delta t + \frac{a(t)\Delta t^2}{2}, \quad (42)$$

where $v(t)$ can be replaced by,

$$v(t) = \frac{x(t) - x(t - \Delta t)}{\Delta t}, \quad (43)$$

The positions of all masses have to be given for the first two time steps in order to run the integration algorithm. These positions can be determined from the initial conditions and for example a simple Euler forward integration in the first time step.

As mentioned above the real-life equivalent of this model is a gas filled tube with rigid walls. Initial and boundary conditions have to be chosen according to the properties of the tube (For example: open/closed ends; excitation over time; initial pressure and velocity distribution.). If the tube has closed ends, the outer masses have to remain at fixed positions. If the tube has open ends, the pressure at the ends is always the ambient pressure therefore the spring force has to stay the same over all time steps. This can be easily achieved by setting the distance between the first and the second mass to a fixed value, whereas the

position of the second mass is always calculated through the time integration scheme.

Excitations can be realized by changing the position of respective masses with time.

The equivalent pressure between each pair of masses can be determined according to Eq. (40) using position information along with the spring force law as given by Eq. (36). In this manner the pressure distribution can be plotted over position at each time step.

Another useful tool for the analysis of acoustic signals is the Fast Fourier Transform (FFT) [18] which represents an efficient implementation of the Discrete Fourier Transform (DFT). The result of this procedure provides discrete magnitude and phase information of the analyzed discrete and time-variant periodic signal. Complicated signals can be dissected into their frequency components, which can then be used to gain additional information about the signal.

In order to obtain information about the system response to excitations or the initial conditions, the time dependent pressure at one point of the model can be studied using a Matlab-based FFT implementation. The lumped mass model includes a visualization of the model motion, a time frame of which is illustrated in Figure 3.3.



Figure 3.3: Visualization of the one-dimensional lumped mass model for an example with nine masses.

The algorithm scales linearly and is perfectly suited for a parallelized implementation because the motion of each mass only depends on the motion of its two neighboring masses.

3.4 Numerical experiments

A series of simulations is carried out to confirm the agreement of simulation results with solutions of linear acoustic wave equation. The parameters employed are provided in Table 3.1.

Table 3.1: Parameters for the Lumped mass models.

tube length, L :	1 m
mean density, ρ_{mean} , of air:	1.2041 kg/m ³
mean atmospheric pressure, p_{mean} :	101325 Pa
isotropic coefficient γ :	1.4
tube cross section area, A : (doesn't affect the results)	1 m ²
time step size:	1e ⁻¹⁷ s

The time step size is chosen to be much smaller than the Courant–Friedrichs–Lewy condition, which is a necessary stability condition. This leads to stable time-marching and accurate results. As mentioned in section 3.1, solutions of the lumped mass model and the linear wave equation are only equivalent if spring forces are linearized. Consequently the agreement of model and linear theory can only be verified if the model draws on the linear spring response as given by Eq. (36). One dimensional tube problems are well suited to verify the model since they have been studied in detail and solutions to various initial and boundary value problems are available in literature [19]. Nonlinear effect can be observed as soon as high pressure amplitudes or frequencies are involved. These effects will also be analyzed in two simulations.

3.4.1 Open ended tube with impulse excitation

In this example a one meter long tube is simulated using the linearized lumped mass model with a discretization of 300 masses. One way to determine the wave speed in the model is to analyze the eigenfrequencies of the system. Open ended tubes only support frequencies of wavelength $\lambda = 2Ln$ where $n \in \mathbb{N}^*$. Using the relation between wavelength and frequency, $\lambda = \frac{c}{f}$, the wave speed, c , can be determined from the first eigenfrequency, f_1 .

$$c = 2Lf_1 \quad (44)$$

An initial pressure impulse can be used to ensure that all frequencies are excited. This can be realized by simply shortening the initial length of one spring in the system. In the following example this impulse is created at a randomly chosen position of $x = 33.3\text{ cm}$. Afterwards the system dynamics is simulated over a total time of $T = 0.5\text{ s}$ and the pressure at position $x = 3.3\text{ cm}$ is analyzed via FFT.

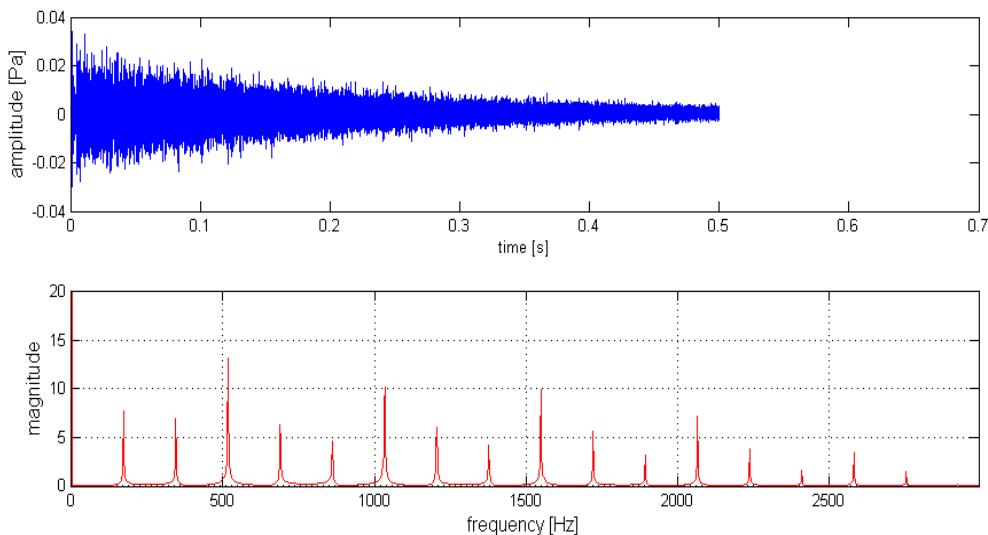


Figure 3.4: Amplitude plot of the pressure signal (top) and FFT-magnitude plot of the same signal (bottom).

The first peak in the magnitude plot can be found at a frequency of $f_1 = 172 \text{ Hz}$, whereas the frequency resolution of the FFT is $1/T = 2 \text{ Hz}$. Using Eq. (44) the wave propagation speed can be determined to $c = 344 \pm 4 \frac{\text{m}}{\text{s}}$. The theoretical propagation speed for ideal gas,

$$c = \sqrt{\gamma \cdot \frac{p_{mean}}{\rho_{mean}}} = 343.23 \frac{\text{m}}{\text{s}}, \quad (45)$$

shows good agreement with the calculated value.

The contour of the spectrum depends on excitation position and also on the position where the pressure is analyzed. The amplitude of the signal becomes smaller with time because the damping forces dissipate energy. This experiment also shows that the chosen boundary formulation models open tube ends accurately. The same wave speed was observed in an experiment with closed tube ends, represented by fixed outer masses.

3.4.2 Triangular initial pressure contour

In this example the initial displacements of the masses have been set to values that result in a triangular initial pressure contour. Linearized spring forces are used again since the simulation results will be compared to an analytic solution of the wave equation which exists for the given initial condition and an open ended tube. The air in the tube is discretized by 700 lumped masses and the dynamics is simulated over a time of $T = 0.2 \text{ s}$. The pressure plots for different times are shown in Figure 3.5.

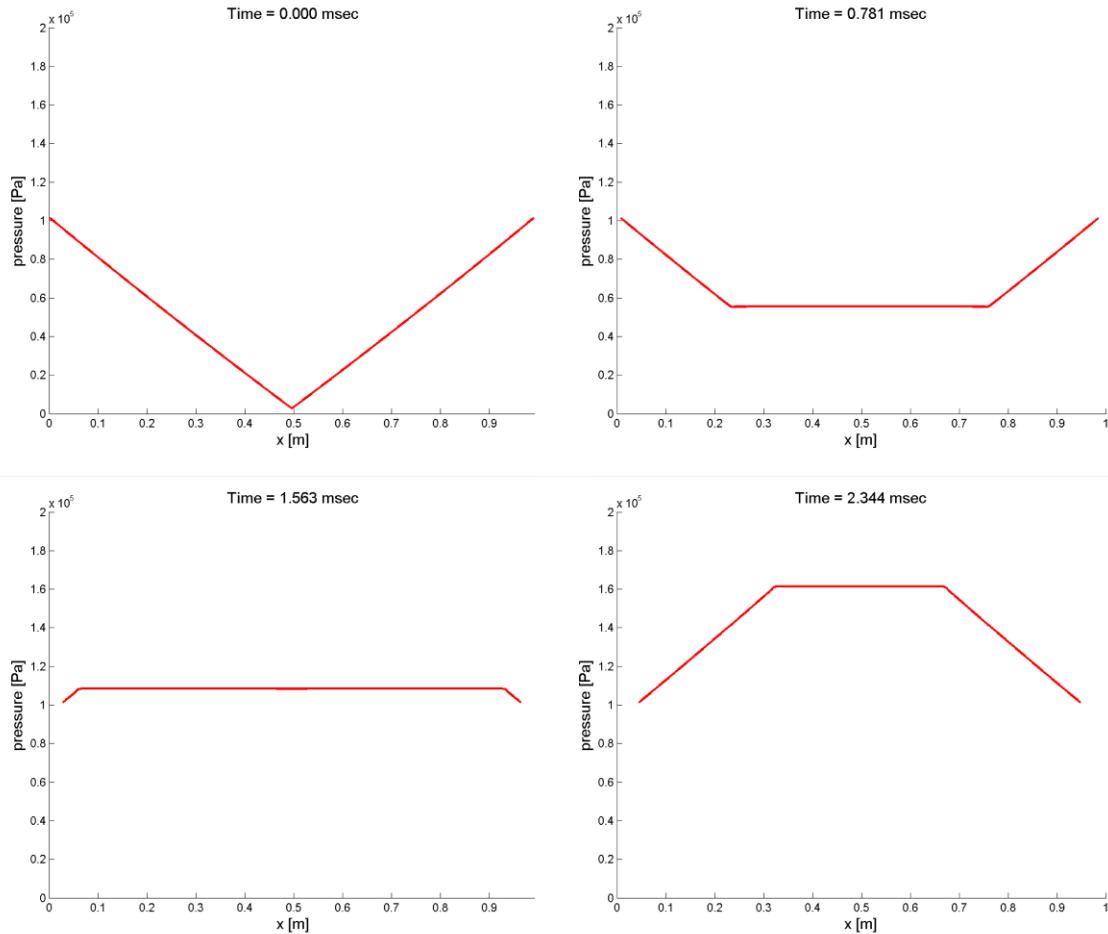


Figure 3.5: Pressure contour recorded at four different times (given at the top of each plot). The initial pressure distribution has a triangular shape.

The results agree well with the analytic solution. Figure 3.6 shows the pressure signal, analyzed between the first two masses of the array (position $x = 7.1e^{-4} m$).

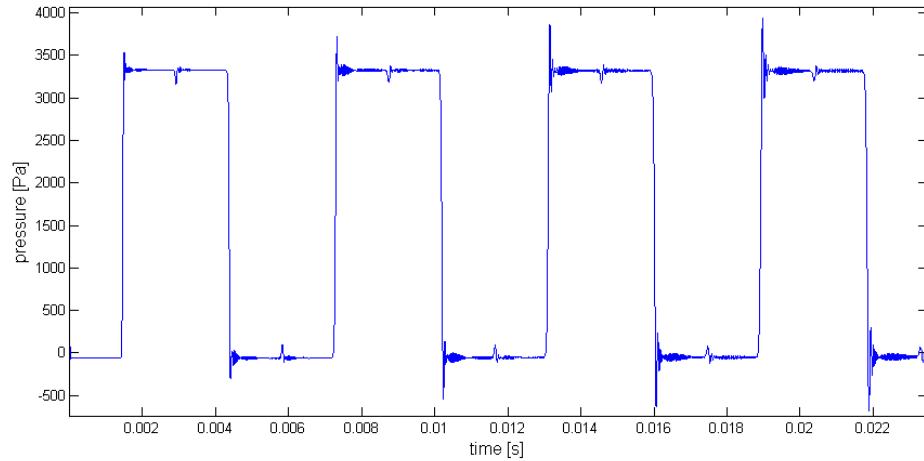


Figure 3.6: Pressure between the first two masses of the lumped mass model plotted over time.

The amplitude of the pressure shows small deviations from the theoretical solution which is a rectangular signal. This is caused by the finite discretization used in the simulation. It can be noticed that the deviations increase over time.

3.4.3 Nonlinear examples

After the linearized lumped mass model showed good agreement with analytic solutions in the previous examples it can be assumed that it is able to model the linear acoustic wave equation accurately. The following two simulations will employ the nonlinear spring formulation according to Eq. (35). As expected the nonlinear model gives virtually the same results as reported above if pressure amplitudes are small. A tube with closed ends and sinusoidal velocity excitation from the right side is chosen to illustrates differences between the linearized and the nonlinear model. The results of the linearized lumped mass model is given in Figure 3.7 at different times.

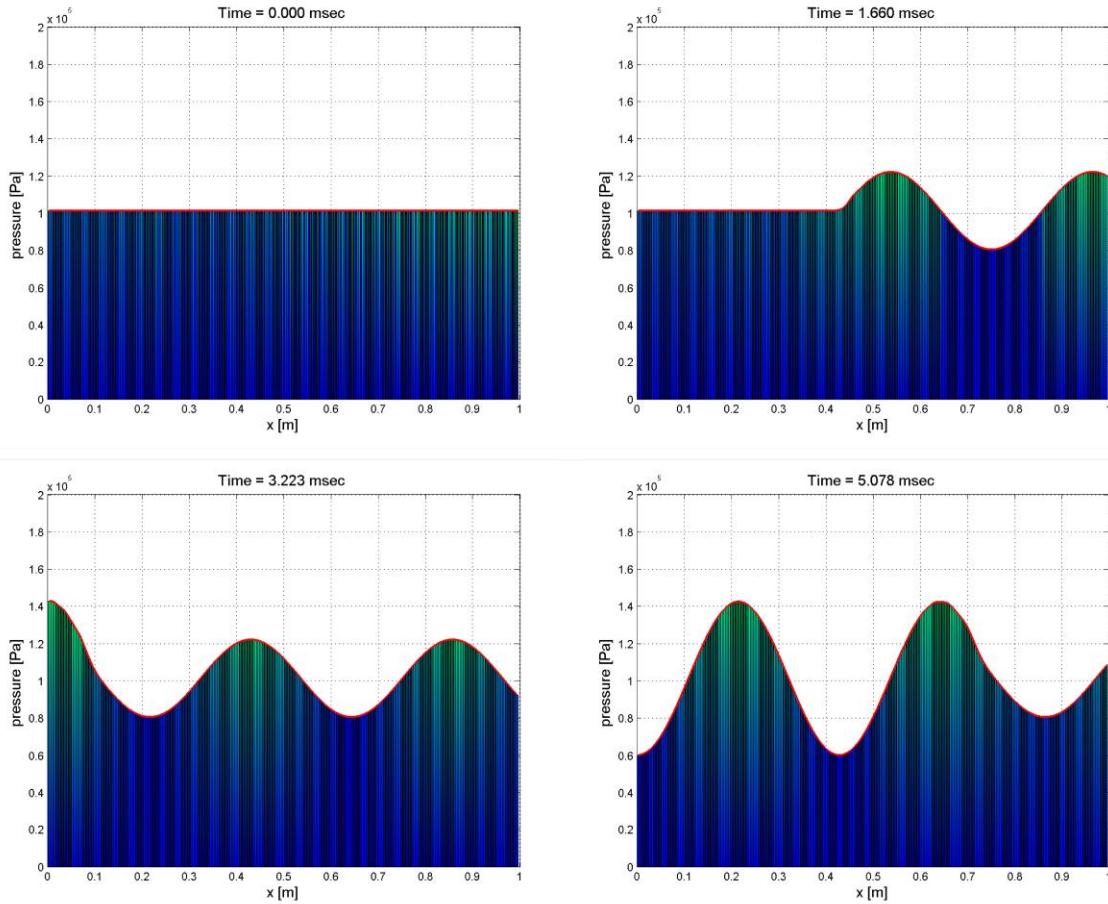


Figure 3.7: Pressure contours simulated with the linearized lumped mass model given. The four pressure graphs are given at different times for zero initial conditions and sinusoidal excitation from the right side.

It can be observed that a sine wave travels through the tube starting at the right end. It gets reflected at the closed left end, which is modeled by a fixed outer mass. The results of the same experiment conducted with the nonlinear model are illustrated in Figure 3.8.

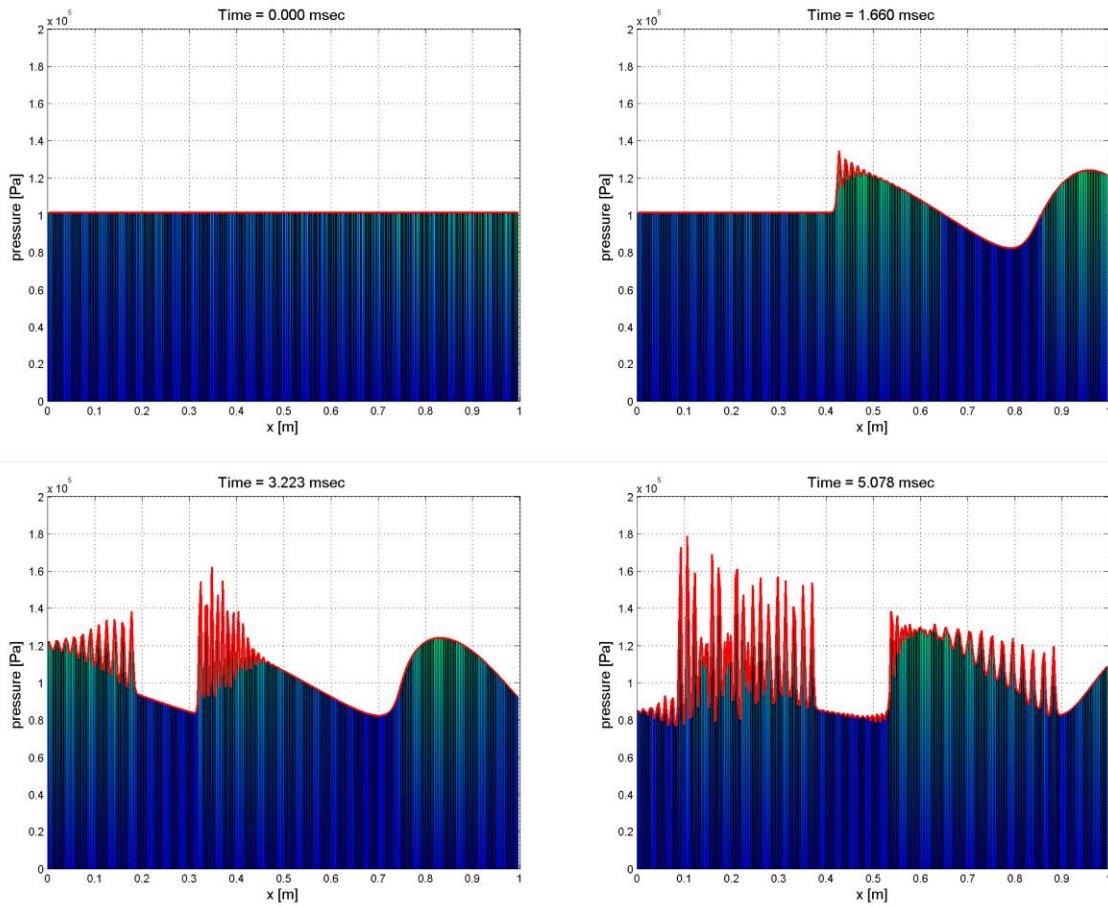


Figure 3.8: Pressure contours simulated with the nonlinear lumped mass model given. The four pressure graphs are given at different times for zero initial conditions and sinusoidal excitation form the right side.

Looking at the pressure plot for time equal to 1.660 ms in Figure 3.8 it can be noticed that the wave form doesn't have a sinusoidal shape. The sawtooth-like contour is due to the effect of variable wave speed. The wave speed in ideal gases varies with pressure and density and can be calculated using Eq. (45). This effect can be explained on a microscopic level since particles transmitting the wave energy are closer for higher pressures, leading to higher collision probability and increased speed of sound. This affects the wave's frequency structure. In the plain sinusoidal excited wave the peaks travel faster than the troughs, leading first to a sawtooth-like wave structure and eventually to splitting into high frequency peaks.

This implies that frequency components, different from the excitation frequency, are introduced. This phenomenon is typical for nonlinear systems. The results are qualitatively similar to the simulation done by Zabusky and Kruskal [20] as illustrated in Figure 3.9. These results have been obtained using the Korteweg-de Vries equation which will be discussed briefly on the next page.

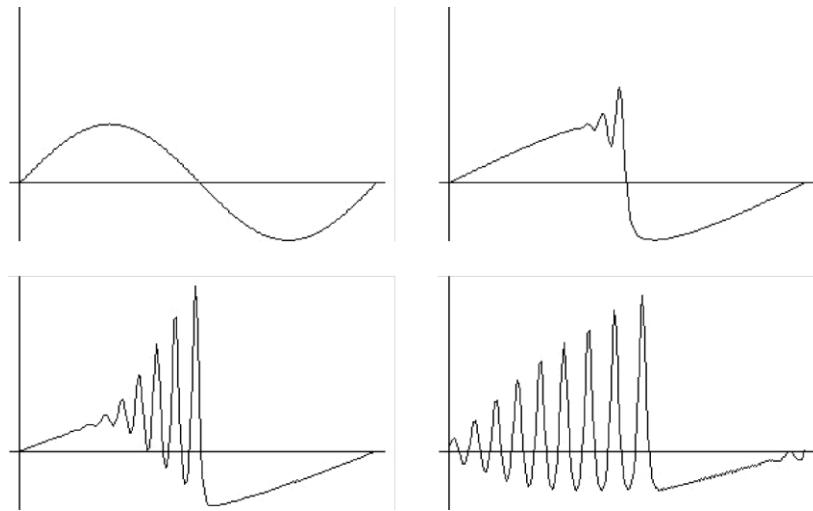


Figure 3.9: Temporal development of an initial sinusoidal wave into a sequence of solitary peaks calculated using the Korteweg-de Vries equation.

A similar effect can be observed in water waves that eventually start to break. However, unlike water waves, acoustic waves propagate purely longitudinally. This explains why no wave breaking occurs in Figure 3.8. The plots for time equal to 3.223 ms and 5.078 ms show that the waves get reflected at the right end of the lumped mass model. Due to the nonlinearity, the waves show interaction as they propagate through each other because of spatially varying propagation speed. Thus the principle of superposition can not be used for this type of system.

Even though it has been shown that sinusoidal waves are not stable in the nonlinear model, it is known that stable acoustic solitary waves called solitons exist [21]. The stability

of solitons results from the balance of nonlinearity and dispersion. Nonlinearity drives a solitary wave to concentrate further whereas dispersion spreads such a localized wave. Thus the shape of solitons is defined by the combination of both the nonlinearity and the dispersion of a physical phenomenon such as sound propagation. In this respect, solitons are completely different from linear waves like the sinusoidal wave in Figure 3.7. In fact, sinusoidal waves are very unstable in nonlinear acoustic simulations and the example above shows that they break up into a sequence of solitons.

Dispersion in acoustics results from material specific microscopic interaction effects that can be described by statistical mechanics. These effects are not captured by a lumped mass model which is only based on laws of continuum mechanics. However, the fact that solitons form in the numerical experiment as shown in Figure 3.8 proves the existence of dispersion in this simulation. This dispersion is caused by the model and the integration algorithm used. Thus, the solitons in Figure 3.8 do not necessarily reflect properties such as shape and width of real acoustic solitons. This is an artifact of all acoustic simulations that are not based on statistical mechanics.

For many nonlinear phenomena solitary waves can be described by the Korteweg-de Vries equation (KdV) [22], and an exact solitary solution of the latter is given by Eq. (46).

$$p(x) = a \cdot \operatorname{sech}^2(b(x - vt - x_0)) \quad (46)$$

It defines the pressure, p , with respect to position x . The wave propagation velocity is called v and the parameters a and b define width and height of the soliton. Different combinations of parameters have been tested but none of the created pressure waves was exactly stable. Therefore the shape of one soliton created in the previous example was isolated and used to

define initial conditions for the next simulation. Solitary waves have the interesting characteristic that they travel with different speeds dependent on their amplitudes and they can run through each other without interactions. In the following example, three solitons of different amplitudes are created using one soliton from the last simulation to define the initial conditions for displacement and velocity. Figure 3.10 shows eight snapshots of the spatial pressure distribution taken at different times.

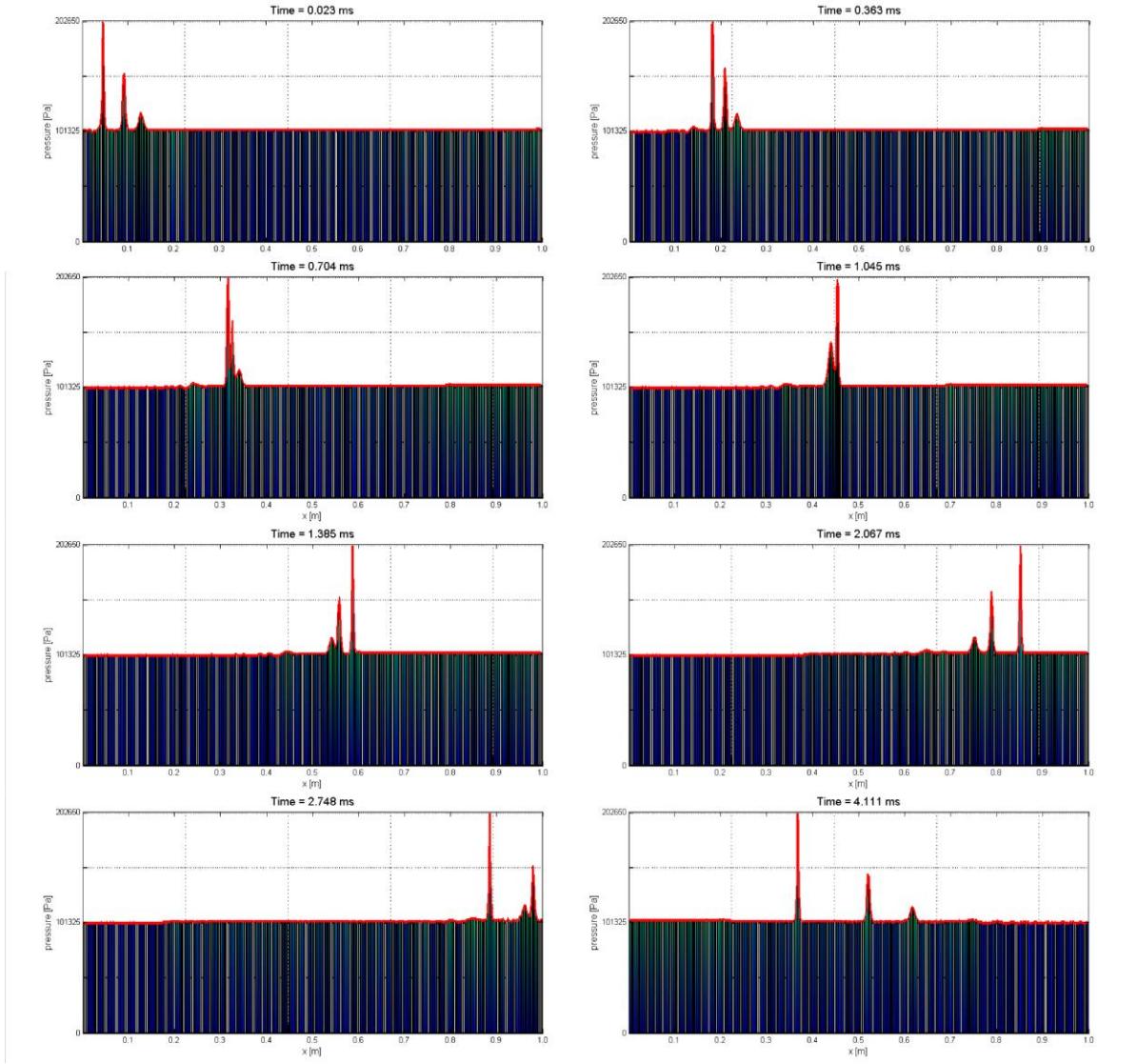


Figure 3.10: Simulation of three soliton waves traveling through a 1 meter long tube with closed ends.
The propagation speed of the solitons increases with amplitude. Due to this effect, higher solitons overtake the smaller ones.

The solitons in Figure 3.10 propagate with different velocities according to the wave height.

The highest peak travels with the highest velocity and overtakes the other waves. Afterwards the gap between the solitons increases until the highest gets reflected from the closed end of the tube. It can be seen that they can run through each other but when this happens the amplitude cannot be calculated using the principle of superposition. The stability of the

waves is sensitive to the discretization, confirming the dependency of soliton stability on the numerical model. This shows that the model is limited to problems where wave diffusion doesn't play an essential role, which is given when pressure gradients are only moderate.

A very challenging problem is the transition of the studied lumped mass model in to two and three dimensions. A cubical mesh of masses that are connected via springs in each direction leads to decoupled pressure values in each direction and is therefore not useful. Random connections would presumably solve this problem but model parameters such as spring stiffness and spring response as well as the maximal connection distance are hard to determine and don't reflect the real physics accurately. For this reason another particle based method called Smoothed Particle Hydrodynamics will be discussed in Chapter 4. This method is well established and provides additional flexibility in comparison to the lumped mass model. However, the idea behind this method is similar to the lumped mass model since the domain is discretized using discrete particles which exert forces on each other.

Chapter 4

Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) was independently developed for simulations in astronomy by Lucy [23] and Gingold & Monaghan [24] in 1977. In SPH, each particle represents a part of the continuum problem domain meaning that a certain amount of fluid is combined into one particle. Depending on the problem, each particle possesses a set of field variables such as mass, density, momentum, or energy. SPH particles have a distance, called smoothing length, associated with them over which their properties are smoothed out over the domain by a kernel function. The time evolution of the system is derived based on the conservation laws for mass, momentum and energy. The functions and their derivatives in these equations are approximated using the field variables defined at the position of the particles in combination with the smoothing kernel. Due to its Lagrangian and meshless formulation, SPH has certain advantages over common mesh based methods that explain its popularity in Hydrodynamics:

- Adaptive nature of the discretization leads to a natural way of handling large deformations.
- No predefined mesh to provide connectivity between particles, which is closer to the physics of most systems.

- Particles carry material properties and are free to move due to external forces and internal interactions, which makes it easy to simulate inhomogeneous problems or moving material interfaces.
- Highly robust and relatively fast solution algorithm.

The following key ideas, which will be discussed in depth in 4.1, are the basis of SPH:

1. The integral representation method is used for the field function approximation. This is known as the kernel approximation in the SPH method. It provides stability as the integral representation has a smoothing effect.
2. The kernel approximation is then further approximated using particles. This is known as the particle approximation in SPH. The integral in the integral representation of the field function and its derivatives are replaced with summations over all the corresponding values at the neighboring particles in a local domain called the support domain (compact support is assumed however it can be instationary (discussed in section 4.6)). This results in a limited number of particle interactions and hence, lower computing times.
3. Particles can be assigned mass and represent physical material.
4. The particle approximations are performed on all terms related to field functions in the PDE's to produce a set of ODE's in discretized form with respect to time only.
5. The ODE's are commonly solved using an explicit integration algorithm to achieve fast time stepping.

4.1 The Basic SPH Formalism

The kernel approximation or integral representation, $\langle f(x) \rangle$, of a field function, $f(x)$, at the spatial position, x , is given by,

$$\langle f(x) \rangle = \int_{\Omega} f(x') W(x - x', h) dx', \quad (47)$$

where the integration is over the entire support domain, Ω , and W is the kernel or smoothing function that can be interpreted as a distance $(x - x')$ dependent weighting function. In the smoothing kernel, h , is the smoothing length defining the influence area of the smoothing function W . The smoothing function is usually chosen to be an even function and it satisfies a number of conditions. Firstly, the normalization or unity condition,

$$\int_{\Omega} W(x - x', h) dx' = 1. \quad (48)$$

Secondly, a Delta function property is observed as the smoothing length approaches zero,

$$\lim_{h \rightarrow 0} W(x - x', h) = \delta(x - x'). \quad (49)$$

The above features assure proper normalization and consistency in the continuum limit. The third condition is the compact condition,

$$W(x - x', h) = 0 \text{ when } |x - x'| > \kappa h, \quad (50)$$

where κ is a constant related to the smoothing function and defines the effective non-zero area of the smoothing function called the support domain. In three dimensions it is a spherical volume of radius κh . This condition also localizes the support domain, Ω , of the smoothing function.

It can be shown, that the kernel approximation corresponding to Eq. (47) has an accuracy of second order in space [25]. In SPH a second approximation is used to convert the

spatial integration to a summation over all particles within the support domain. This is also referred to as the particle approximation. The differential volume, dx' , at the location of particle j is replaced by the finite volume, ΔV_j , of the particle that is related to the particle mass, m_j , by,

$$m_j = \Delta V_j \rho_j, \quad (51)$$

where density, ρ_j , is the mass per unit volume of the particles for $j = 1, 2, \dots, N$ where N is the number of particles in the support domain. The SPH representation for $f(x)$ can be rewritten using the particle approximation as,

$$\begin{aligned} \langle f(x) \rangle &= \int_{\Omega} f(x') W(x - x', h) dx' \\ &\approx \sum_{j=1}^N f(x_j) W(x - x_j, h) \Delta V_j h \\ &= \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) W(x - x_j, h). \end{aligned} \quad (52)$$

Figure 4.1 shows a two dimensional graphical interpretation of the particle approximation.

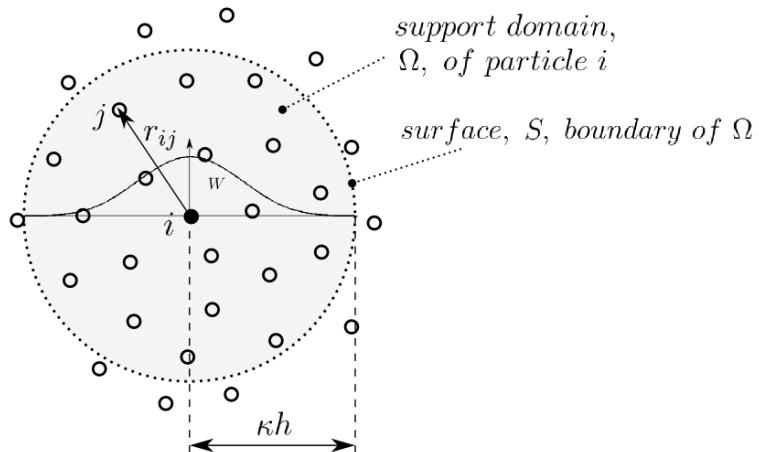


Figure 4.1: This figure illustrates the Kernel, W , and its support domain, Ω , which is bordered by S . The field function at every point (not necessarily at a particle) can be approximated by

summing up the weighted function values at the positions of the particles in the support domain [26].

An essential point is that spatial derivatives of a field function can be constructed from known function values at the particles by using a kernel which is differentiable [26]. There is no need to use finite differences and no need for a grid. The approximation for the spatial derivative, $\nabla f(x)$, is obtained by simply substituting $f(x)$ with $\nabla f(x)$ in Eq. (47),

$$\langle \nabla f(x) \rangle = \int_{\Omega} [\nabla f(x')] W(x - x', h) dx'. \quad (53)$$

The divergence in the preceding integral is on the primed coordinate, moving with the particle. Using the product rule of differentiation, $\nabla(a \cdot b) = \nabla a \cdot b + a \cdot \nabla b$, the integrant can be replaced.

$$[\nabla f(x')] W(x - x', h) = \nabla[f(x') W(x - x', h)] - f(x') \cdot \nabla W(x - x', h) \quad (54)$$

Substituted in Eq. (53) the following expression can be found,

$$\langle \nabla f(x) \rangle = \int_{\Omega} \nabla[f(x') W(x - x', h)] dx' - \int_{\Omega} f(x') \cdot \nabla W(x - x', h) dx'. \quad (55)$$

Using the divergence theorem, $\iiint_{\Omega} (\nabla \cdot \mathbf{F}) d\Omega = \iint_{\partial V} \mathbf{F} \cdot \mathbf{n} dS$, the first integral on the right hand side can be converted into an integral over the surface, S , of the integration domain, Ω .

Thus, Eq. (55) can be replaced by,

$$\langle \nabla f(x) \rangle = \int_S f(x') W(x - x', h) \cdot \mathbf{n} dS - \int_{\Omega} f(x') \cdot \nabla W(x - x', h) dx', \quad (56)$$

where \mathbf{n} is the outwards pointing unit vector normal to the surface. Since the kernel, W , has compact support, the surface integral is zero and vanishes. It should be noted that the dimension of the surface integral is always one less than the problem dimension.

Consequently for a one dimensional problem domain the surface integral is simply equal to the integrand. This fact is used in section 4.7.5 to formulate boundary conditions.

Using the compact condition, as shown in Figure 4.2, Eq. (56) can finally be reduced to the following expression for spatial derivatives of field functions in the domain,

$$\langle \nabla f(x) \rangle = - \int_{\Omega} f(x') \cdot \nabla W(x - x', h) dx'. \quad (57)$$

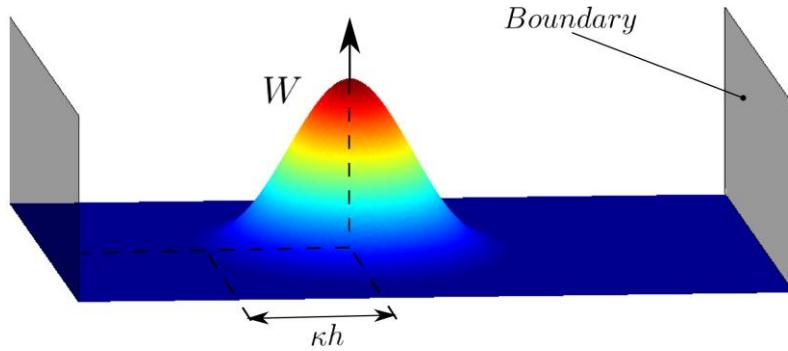


Figure 4.2: This figure shows the kernel function, W , in two dimensions. The surface integral in Eq. (56) is equal to zero as long as the kernel function is compact or in other words equal to zero at the surface or boundary of the support domain.

In the next step the particles approximation according to Eq. (52) is applied on the previous expression which can consequently be written as follows,

$$\langle \nabla f(x) \rangle = - \sum_{j=1}^N \frac{m_j}{\rho_j} f(x_j) \nabla W(x - x_j, h). \quad (58)$$

That is, the spatial gradient of the field function can be calculated from the field function and the derivatives of the smoothing function W , rather than from the derivatives of the field function.

It should be noted that the surface integral in Eq. (56) is only zero if the support domain does not overlap the problem domain boundary as shown in Figure 4.3.

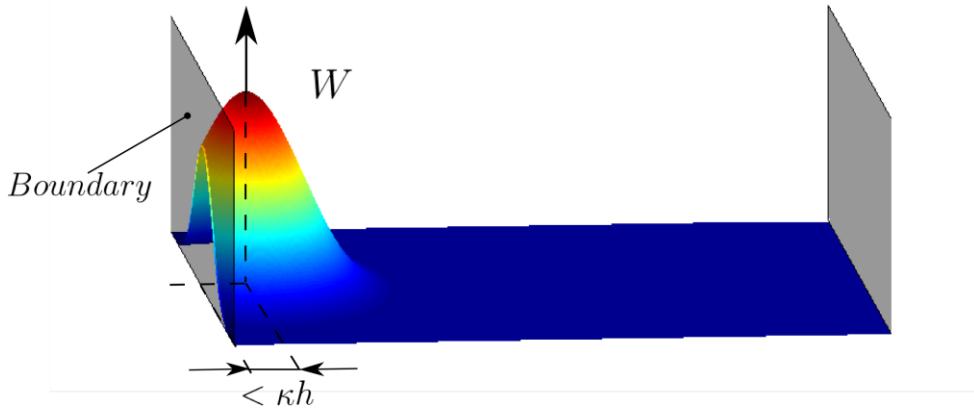


Figure 4.3 Kernel function, W , close to the domain boundary. In case the support domain of the kernel overlaps the boundary of the problem domain the surface integral in Eq. (56) can be non-zero.

This leads to the biggest problem of SPH which is to find a consistent boundary formulation.

Different approaches will be discussed in section 4.7.

The SPH formalism can be summarized as follows:

Particles have a spatial distance, κh , associated with them over which their properties are "smoothed" by the smoothing kernel. This means that the field function at any point can be obtained by summing the weighted field function values of all the particles which lie within the range of the kernel. The contributions of each particle to the field function at the point of interest are weighted according to their distance from the point, their mass and their density. Mathematically, this is governed by a summation according to Eq. (52). Spatial derivatives can be calculated similarly using field function values known at particle positions.

4.2 SPH Formulation for Fluid Dynamics and Acoustic Simulations

The SPH formulation for fluid dynamics and acoustics is derived by spatially discretizing the continuity equations of continuum mechanics in Lagrangian form, specifically these are:

$$1) \text{ Conservation of mass:} \quad \frac{d\rho}{dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta} \quad (59)$$

$$2) \text{ Conservation of momentum:} \quad \frac{dv^\alpha}{dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} + f^\alpha \quad (60)$$

$$3) \text{ Conservation of energy:} \quad \frac{du}{dt} = \frac{\sigma^{\alpha\beta}}{\rho} \frac{\partial v^\alpha}{\partial x^\beta} \quad (61)$$

where α denotes the coordinate direction and β stands for repeated indices. As already mentioned, ρ stands for density, v for velocity, σ for stress, u for internal energy and f^α for body forces. For Newtonian fluids, the total stress tensor, $\sigma^{\alpha\beta}$, consists of the isotropic pressure, p , and the viscous stress $\sigma_{visc}^{\alpha\beta}$,

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \sigma_{visc}^{\alpha\beta}. \quad (62)$$

This notation uses $\delta^{\alpha\beta} = 1$ if $\alpha = \beta$ and $\delta^{\alpha\beta} = 0$ otherwise. The SPH formulation of the preceding equations leads to a set of ordinary differential equations (ODEs) which can then be solved via numerical time integration. The general SPH formulation for dynamical flow also includes artificial viscosity and heat for shock problems.

4.2.1 Conservation of mass

The intuitive way to calculate density is to simply replace f in Eq. (52) with ρ which results in the following expression,

$$\langle \rho_i \rangle = - \sum_j \rho_j W_{ij} . \quad (63)$$

The summation is carried out over all particles, j , in the support domain. It has been shown by Monaghan [27] that an alternative expression, given by,

$$\frac{d\rho_i}{dt} = \sum_j m_j (v_i - v_j) \nabla_i W_{ij} , \quad (64)$$

which is based on the continuity Eq. (59) and expressed in the form of Eq. (58), has certain advantages. In this notation, ∇_i is the Lagrangian. The expression leads to higher accuracy for problems involving free-boundaries and also establishes a computational advantage since the rate of change of all the field variables can be calculated in one loop. On the other hand Eq. (64) does not enforce exact mass conservation. It has to be evaluated which representation of the continuity equation (Eq. (63) or Eq. (64)) is more advantageous for a specific application.

4.2.2 Conservation of momentum

The equations of motion will first be formulated without any viscous forces but, as described later, these forces are introduced using artificial viscosity. The resulting simplified momentum equation is called the Euler Equation of fluid dynamics and has form,

$$\frac{dv}{dt} = -\frac{1}{\rho} \nabla p . \quad (65)$$

Using Eq. (58) it can be written as follows,

$$\frac{dv_i}{dt} = -\frac{1}{\rho_i} \sum_j m_j \frac{p_j}{\rho_j} \nabla_i W_{ij}, \quad (66)$$

where the total stress tensor, $\sigma^{\alpha\beta}$, according to Eq. (62) was already replaced by the isotropic pressure, p . The summation is again over all particles, j , in the support domain. However Eq. (66) does not conserve linear and angular momentum exactly because the force on particle i due to particle j is not equal to the force on particle j due to particle i . Using the identity,

$$\frac{\nabla p}{\rho} = \nabla \left(\frac{p}{\rho} \right) + \frac{p}{\rho^2} \nabla \rho, \quad (67)$$

and applying the SPH formulation according to Eq. (58) on the two expressions on the right hand side leads to a symmetric expression for the equation of motion.

$$\frac{dv_i}{dt} = - \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} \right) \nabla_i W_{ij} \quad (68)$$

Viscosity effects can be introduced either directly in the momentum equation [28] or indirectly with the so called artificial viscosity. The most widely used expression for artificial viscosity in SPH simulations was introduced by Monaghan and Gingold [29]. It has the advantage of containing a term linear in velocity, which produces a representation of the classical shear and bulk viscosities, as well as a quadratic velocity term which handles high Mach number shocks. The discretized equation of motion, Eq. (68), is simply expanded with the artificial viscosity term, Π_{ij} , which can be interpreted as an additional pressure replacing viscous forces.

$$\frac{dv_i}{dt} = - \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) \nabla_i W_{ij} \quad (69)$$

The artificial viscosity is given by the following expression,

$$\Pi_{ij} = -\nu \left(\frac{v_{ij} \cdot r_{ij}}{r_{ij}^2 + \epsilon \bar{h}_{ij}^2} \right) \quad (70)$$

$$\nu = \frac{\bar{h}_{ij}}{\bar{\rho}_{ij}} \left(\alpha \bar{c}_{ij} - \beta \frac{\bar{h}_{ij} v_{ij} \cdot r_{ij}}{r_{ij}^2 + \epsilon \bar{h}_{ij}^2} \right), \quad (71)$$

where $\bar{h}_{ij} = (h_i + h_j)/2$ is the mean smoothing length, $v_{ij} = v_i - v_j$ is the relative velocity, r_{ij} is the particle distance and $\bar{\rho}_{ij} = (\rho_i + \rho_j)/2$ is the mean density of the interacting particles i and j . The parameters α and β have to be adjusted for every application, and ϵ can be set to 0.01 in order to prevent singularities when r_{ij} approaches zero.

Eq. (69) in combination with Eqs. (70) and (71) successfully reproduce the linear shear and bulk viscosity of fluids and was also found to stabilize shocks. SPH particles can also unphysically stream “through” each other which mostly occurs in simulations of shock waves. Artificial viscosity provides a solution to this problem since it converts kinetic energy to internal energy which causes a pressure gradient and therefore a repulsive force preventing the particles from steaming through each other.

When simulating high Mach number compressible gas flows where the real viscosity is negligible, then Eq. (70) only applies when the fluid is being compressed, i.e. when particles are approaching one another, which is fulfilled when $v_{ij} \cdot r_{ij} < 0$. If the gas is expanding, $v_{ij} \cdot r_{ij} > 0$, the artificial viscosity can be set to zero, $\Pi_{ij} = 0$. In many high speed flow applications it has been found that the choice of $\alpha = 1$ and $\beta = 2$ produces excellent results.

At the opposite end of the spectrum, at low Mach number flows, where the fluid has significant real viscosity, Eq. (71) can be used with $\beta = 0$. The value of α is then chosen so that the simulated fluid has a viscosity and Reynolds number similar to that of the real fluid being modeled. Values are given in literature for different liquids and gases [30]. By

performing a more careful analysis of the artificial viscosity equation Monaghan and Kos [31] have shown that it leads to a shear viscosity of the form,

$$\mu = \frac{1}{8} \alpha h c \rho \quad (72)$$

Monaghan [30] has used equation (71) with $\beta = 0$ and $\alpha = 0.01$ to simulate the evolution of an elliptical water drop, the bursting of a dam, and the formation of a bore. In each case excellent agreement was found between with experimental and simulation results. The run-up and return of a solitary wave travelling over shallow water and then onto a dry beach backed by a vertical wall was simulated by Monaghan and Kos [31] using the parameters $\beta = 0$ and $\alpha = 0.001$. The simulations accurately reproduced experimental results. Whilst the use of artificial viscosity has been shown to give accurate results in many scenarios, difficulties have been observed in some special cases, for example when simulating highly viscous fluids.

For SPH simulations involving two or more fluids with very different viscosity values Cleary has used a more sophisticated viscous term, which automatically ensures that the stress is continuous across material interfaces [32]. Cleary et al. [33] have shown that this form of the viscosity allows multiple materials with densities and viscosities varying by up to three orders of magnitude to be accurately simulated.

4.2.3 Conservation of energy

The conservation laws above do not require the rate of change of internal energy to be calculated. However, it is convenient to convert the non-dissipative rate of change of internal energy to its SPH form. Then, it becomes possible to account for pressure rises due to

temperature changes. From the first law of thermodynamics per unit mass, (T is temperature),

$$\begin{aligned} Tds &= du + pdv \\ &= du - \frac{p}{\rho^2} d\rho, \end{aligned} \quad (73)$$

and under the assumption of an adiabatic process, the rate of change of the specific internal energy can be written as follows,

$$\frac{du}{dt} = \frac{p}{\rho^2} \frac{d\rho}{dt} = -\frac{p}{\rho^2} \nabla v, \quad (74)$$

whereas in the second expression, $d\rho/dt$, was replaced according to the conservation of mass.

The SPH formulation of this equation for particle i can be expressed according to Eq. (58) as,

$$\frac{du_i}{dt} = \frac{p_i}{\rho_i^2} \sum_j m_j v_{ij} \nabla_i W_{ij}. \quad (75)$$

Similar to the SPH formulation of the conservation of momentum, Eq. (75) can be expressed in a symmetric form,

$$\frac{du_i}{dt} = \frac{1}{2} \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} \right) v_{ij} \nabla_i W_{ij}. \quad (76)$$

Viscosity dissipates kinetic to internal energy. Owing to the way in which the SPH viscosity was derived, viscous dissipation is best obtained directly from the SPH equations. According to [34] the final thermal energy equation reads,

$$\frac{du_i}{dt} = \frac{1}{2} \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) v_{ij} \nabla_i W_{ij}. \quad (77)$$

Herein, Π_{ij} is the same artificial viscosity term as used in Eq. (69).

4.2.4 Equation of State

The set of conservation equations, Eqs. (64),(69) and (77), with the constitutive equation, Eq. (62), collected in the previous section is still not closed. A closure relationship which relates the hydrostatic pressure to the local density is needed. One advantage of SPH is that the equation of state can be as complicated as desired and it is possible to include state transitions, making SPH very flexible in this context.

4.2.4.1 Ideal Gas

For gases, the most common expression of the equation of state can be derived from the ideal gas law, given in specific form.

$$pV = mRT \quad (78)$$

Herein, R is the specific gas constant and V is the total volume. For an ideal gas the internal energy can be expressed as follows, where γ is the ratio of specific heats,

$$u = c_v T = \frac{RT}{\gamma - 1}. \quad (79)$$

Solving Eq. (79) for T and substituting in Eq. (78) leads to the formulation of the equation of state which relates the pressure, p , to the density, ρ , and the internal energy, u .

$$p = (\gamma - 1)\rho u \quad (80)$$

The isentropic expansion factor, γ , can be determined from the degrees of freedom and is equal to 5/3 for monatomic gas and 7/5 for diatomic gas.

4.2.4.2 Incompressible Liquids

Since Chapter 5 reports on SPH numerical experiments dealing with liquids the commonly used equation of state for incompressible liquids will also be provided subsequently. A realistic equation of state for incompressible liquids is, due to the necessary small time stepping, computationally too expensive to solve. Therefore, a simplified substitute is used herein. In SPH simulations the real fluid is approximated by an artificial fluid that is more compressible than the real fluid. As long as the speed of sound is much larger than the speed of the bulk flow, the artificial fluid will still provide a valid approximation to the motion of the incompressible fluid. Density fluctuations can be limited to $\sim 1\%$ provided the Mach number $M \lesssim 0.1$. This means that the standard SPH formulation described above can still be used. The only modification required is to change the equation of state of the nearly incompressible fluid so that the material becomes more compressible, while retaining sufficient incompressibility so that the speed of sound is large enough to keep the relative density fluctuations small. A widely used equation of state for water, in hydrodynamic simulations, was introduced by Cole [35],

$$p = B \left(\left(\frac{\rho}{\rho_0} \right)^n - 1 \right), \quad (81)$$

where p and the constant B are measured in atmospheres and ρ_0 is the density at atmospheric pressure. The parameter n is usually set to $n = 7$. Thus the only modification required is to change the constant B according to the simulated flow. With the equation of state given by Eq. (81) the speed of sound, c , at reference density, ρ_0 , is given as,

$$c^2 = \frac{nB}{\rho_0}. \quad (82)$$

Therefore, if $B = 100\rho_0v^2/n$, the relative density fluctuations should be $\lesssim 0.01$. To choose the value of B for a specific problem, an upper bound on the speed of the fluid should be known apriori.

4.3 Particle motion

The motion of SPH particles obeys the following equation,

$$\frac{dx_i}{dt} = v_i, \quad (83)$$

where x_i indicates the position of particle i and v_i is calculated using Eq. (69). For nearly incompressible fluids it was observed that the use of Eq. (83) can lead to instabilities. These instabilities can be overcome with a slightly different formulation that takes into account the motion of neighbor particles. This formulation is called XSPH (from extended SPH) and was introduced by Monaghan [27],

$$\frac{x_i}{dt} = v_i + \epsilon \sum_j \frac{m_j}{\rho_{ij}} (v_j - v_i) W_{ij}, \quad (84)$$

where $\rho_{ij} = (\rho_i + \rho_j)/2$ and ϵ is a factor which should have a value between 0 and 1 (usually 0.1). The correction term in equation (84) represents the difference between the velocity of the i -th particle and the mean flow. It has been shown that this expression still conserves both linear and angular momentum and prevents penetration between the particles. This smoothing procedure has proven to reduce the fluctuations present in the velocity field and stabilize the code. However for the numerical experiments discussed in Chapter 6 the standard form given by Eq. (83) was applied.

4.4 Time integration

A popular technique for the time integration of the ordinary differential equations (Eqs. (64), (69), (77) and (83)) is the Leapfrog algorithm which is very efficient and uses little memory [36].

In this second order integration method, the particle velocities and positions are offset by half a time step. At the end of the first time step, t_0 , the changes in field functions are used to update the field functions by half a time step, while the particle positions are advanced a full time step.

$$\begin{cases} t = t_0 + \Delta t \\ \rho_i(t_0 + \Delta t/2) = \rho_i(t_0) + \frac{\Delta t}{2} \frac{\partial}{\partial t} \rho_i(t_0) \\ v_i(t_0 + \Delta t/2) = v_i(t_0) + \frac{\Delta t}{2} \frac{\partial}{\partial t} v_i(t_0) \\ u_i(t_0 + \Delta t/2) = u_i(t_0) + \frac{\Delta t}{2} \frac{\partial}{\partial t} u_i(t_0) \\ x_i(t_0 + \Delta t) = x_i(t_0) + \Delta t \cdot v_i(t_0 + \Delta t/2) \end{cases} \quad (85)$$

At the beginning of each time step the density, velocity and energy of each particle needs to be predicted at half a time step to get a better approximation.

$$\begin{cases} \rho_i(t) = \rho_i(t - \Delta t/2) + \frac{\Delta t}{2} \frac{\partial}{\partial t} \rho_i(t - \Delta t) \\ v_i(t) = v_i(t - \Delta t/2) + \frac{\Delta t}{2} \frac{\partial}{\partial t} v_i(t - \Delta t) \\ u_i(t) = u_i(t - \Delta t/2) + \frac{\Delta t}{2} \frac{\partial}{\partial t} u_i(t - \Delta t) \end{cases} \quad (86)$$

At the end of each subsequent time step, the density, energy, velocity and position are advanced in the standard leapfrog scheme.

$$\begin{cases} t = t + \Delta t \\ \rho_i(t + \Delta t/2) = \rho_i(t - \Delta t/2) + \Delta t \frac{\partial}{\partial t} \rho_i(t) \\ v_i(t + \Delta t/2) = v_i(t - \Delta t/2) + \Delta t \frac{\partial}{\partial t} v_i(t) \\ u_i(t + \Delta t/2) = u_i(t - \Delta t/2) + \Delta t \frac{\partial}{\partial t} u_i(t) \\ x_i(t + \Delta t) = x_i(t) + \Delta t \cdot v_i(t + \Delta t/2) \end{cases} \quad (87)$$

The time step Δt is limited by the Courant–Friedrichs–Lowy condition [37]. In SPH terms, this becomes $\Delta t \leq h/c$. Moreover, if viscosity is present this leads to an additional diffusive limitation on the time step and the two effects are usually combined in the following expression,

$$\Delta t_{ev} = \min_a \left[\frac{h}{c + 0.6 (\alpha c + \beta \max_b \mu_{ab})} \right]. \quad (88)$$

A further limitation on the time step applies if external body forces are present. This implies that the time step should be less than Δt_f , where $\Delta t_f = (h/|f|)^{1/2}$ and f is the external force per unit mass on each of the SPH particles. A suitable time step for the scheme needs to be chosen smaller in order to increase robustness. A good time step size has the form,

$$\Delta t = \frac{1}{4} \min(\Delta t_{ev}, \Delta t_f). \quad (89)$$

4.5 The Smoothing Kernel

The kernel function or sometimes called smoothing kernel is important not only because it defines the shape of the field function approximation and the effective support domain, but it also determines the accuracy of both the kernel and the particle approximation. Kernel functions commonly used include the Gaussian function and the cubic spline. The original calculations of Gingold & Monaghan [34] used a Gaussian kernel in a form that mimics a delta function when $h \rightarrow 0$. The Gaussian smoothing kernel is given by,

$$W(R) = \alpha_d e^{-R^2}, \text{ where } R = \frac{r}{h} = \frac{|x - x'|}{h}. \quad (90)$$

The parameter α_d is $1/(\pi^{1/2}h)$, $1/(\pi h^2)$ and $1/(\pi^{3/2}h^3)$ in 1-, 2- and 3-D respectively.

A kernel based on splines (Monaghan & Lattanzio [38]) has proven computationally efficient, yet, for the physical consistency of SPH equations, it is always best to assume the kernel to be Gaussian.

Other than the Gaussian kernel, the cubic spline kernel is exactly zero for particles further away than two smoothing lengths. This has the advantage of saving computational effort by not including the relatively minor contributions of distant particles. The cubic spline or B-spline is given by,

$$W(R) = \alpha_d \cdot \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & 0 \leq R \leq 1 \\ \frac{1}{6}(2-R)^3 & 1 \leq R \leq 2 \\ 0 & R \geq 2 \end{cases}, \quad (91)$$

where α_d is $1/h$, $15/(7\pi h^2)$ and $3/(2\pi h^3)$ in 1-, 2- and 3-D respectively. There are several other kernel functions that can be constructed in order to fulfill the specific requirements of every simulation. However, all simulation results, reported in Chapter 6, draw on the cubic spline of Eq. (91).

4.6 Variable Smoothing Length

Although the size of the smoothing length can be fixed in both space and time, this does not take advantage of the full power of SPH. By assigning each particle its own smoothing length and allowing it to vary with time, the resolution of a simulation can be made to automatically adapt itself depending on local conditions. This is especially useful in simulations which include areas of very high and very low density. For example, in a very dense region where many particles are close together, the smoothing length can be made

relatively short, yielding high spatial resolution. Conversely, in low-density regions where individual particles are far apart and the resolution is low, the smoothing length can be increased, optimizing the computation for the regions of interest. If the pressure and density variations in an acoustic SPH simulation are small, there is no need to change the smoothing length. In other applications, for instance in blast simulations, the density changes can be significant. In these cases it is advantageous to employ an adaptive smoothing length.

One way to adapt the smoothing length according to the density of the particle is given in Eq. (92) as,

$$h_i = \sigma \left(\frac{m_i}{\rho_i} \right)^{1/d}. \quad (92)$$

Herein, d is the number of dimensions of the simulation and σ is a constant (usually $\simeq 1.3$). For the simulation of nearly incompressible fluids, it is sufficient to choose a fixed smoothing length since the density does not change substantially. The smoothing length represents the effective width of the kernel and its value determines the number of particles with which a given particle interacts. The accuracy of an SPH simulation depends on having a sufficient number of particles within the smoothing length to ensure that the replacement of an integral by a summation is valid. The speed of the computation however decreases as the number of such particles increases. The optimum number of particles has been discussed by Morris [39] and depends on the number of dimensions in the problem. In one, two and three-dimensional problems these are approximately 5, 21 and 57 respectively.

4.7 Boundary Formulations

The simulation of fluid flow or acoustics typically involves fixed or moving boundaries that represent physical interfaces. Problems arise from so called “particle deficiency” at boundaries (see Figure 4.3) leading to erroneous particle approximation of domain integrals (see Eq. (52)). Unlike free surfaces, which are automatically satisfied in SPH, solid boundaries require special care. As already mentioned, boundary formulations are a major issue in SPH simulations. Hence, various approaches to this problem have been proposed in literature but none of them has been found to remedy all issues associated with boundaries. For this reason many approaches are currently used and no boundary formulation has attained general acceptance yet. Four different boundary formulations, used in Chapter 6 for acoustic SPH simulations, are described in the following section.

4.7.1 Dynamic Boundary Particles

The most basic boundary formulation is given in the form of dynamic boundary particles, meaning that SPH particles are used to discretize walls or other obstacles. The conservation laws for mass and energy are evaluated in the usual way and the only difference from fluid particles is that the boundary particles remain fixed or move according to some externally imposed function. When a fluid particle approaches a boundary, the density and the internal energy of boundary particles increase what is resulting in a pressure increase (see Eq. (80)). Thus repulsive forces are created due to the pressure term in the momentum equation (see Eq. (69)). In this way, fixed or moving boundary interfaces can be modeled easily. In order to prevent penetration, several layers of these boundary particles have to be

used. This leads to a boundary layer that has a typical thickness of approximately twice the smoothing length, $2h$.

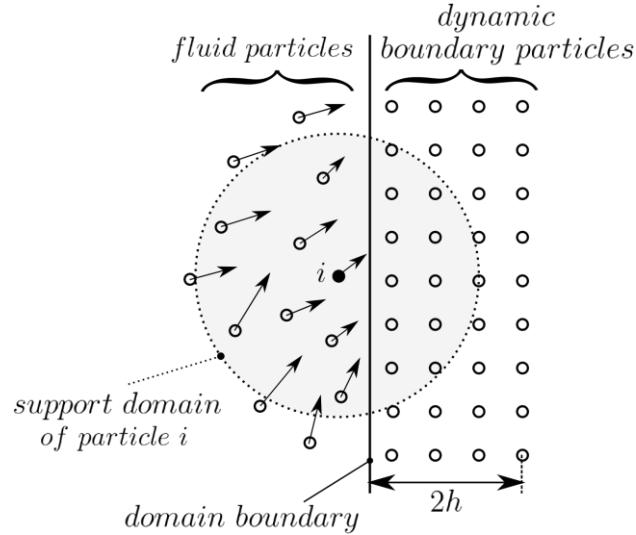


Figure 4.4: Dynamic boundary particles placed outside the domain to remove boundary deficiency. The thickness of this boundary particle layer has to be twice the smoothing length. Unlike fluid particles, dynamic boundary particles do not move relative to the boundary.

But yet, for high velocity flows, the fluid particles can still penetrate boundaries or get stuck in the boundary layer. Additionally, it has been observed that dynamic boundary particles lead to disturbances as fluid particles move along the fluid-boundary interface. Consequently, dynamic boundary particles should only be used for low velocity flows with no tangential flow at boundaries.

4.7.2 Repulsive Forces

Another method to prevent boundary penetration is to create artificial repulsion forces at the boundaries. Typically, the Lennard-Jones potential is applied to model the force acting between boundary and fluid particles. The most obvious way to specify the direction of the force would be the direct connection between the centers of the fluid and boundary particles.

However, high force variations on fluid particles moving parallel to a boundary create large disturbances. This effect is due to the changing fluid particle - boundary particle distance as the fluid particle moves along a boundary. An improved procedure, which doesn't create disturbances, was proposed by Monaghan [40] and successfully applied for hydrodynamic dynamic problems with complicated shaped boundaries. In this formulation, forces are always directed perpendicular to the surface and their absolute value is determined on the basis of the surface normal distance.

4.7.3 Mirroring Particles

Liberky and Petschek first introduced a method for the handling of SPH boundary conditions by relying on the symmetrical mirroring of fluid particles at the boundary surface [41]. This method is called "mirror particles" or "ghost particles". As illustrated in Figure 4.5 if a fluid particle is located within a distance of $2h$ from the boundary a mirror particle is placed symmetrically outside of the domain.

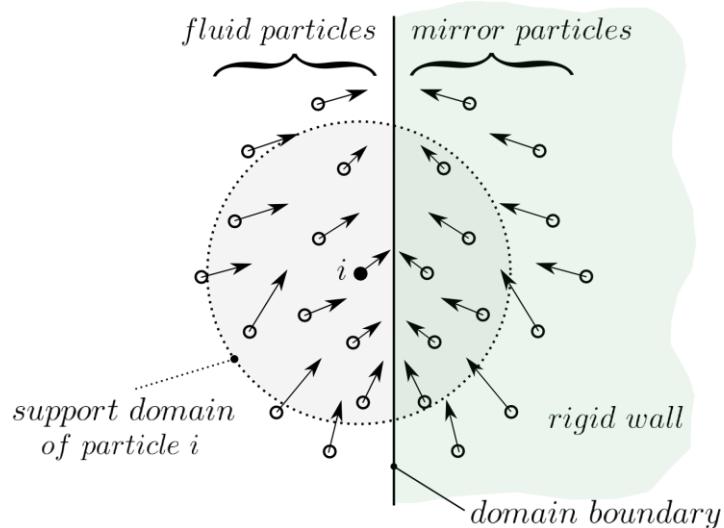


Figure 4.5: Location and velocity of mirror particles. The support domain of particle i overlaps the domain boundary and potential particle deficiency is avoided through mirror particles.

These mirror particles have the same density and internal energy values as their corresponding fluid particles only the component of the velocity perpendicular to the boundary is opposite. However, the velocity of the ghost particles can be chosen in a variety of ways, expressing different boundary conditions. For instance, they can also have opposite velocity, leading to a no slip boundary condition. This boundary formulation seems to reflect boundary conditions exactly but it suffers from the particle approximation. If the boundary is curved or has a sharp edge, the mirroring technique can not completely remove particle deficiency (see Figure 4.6).

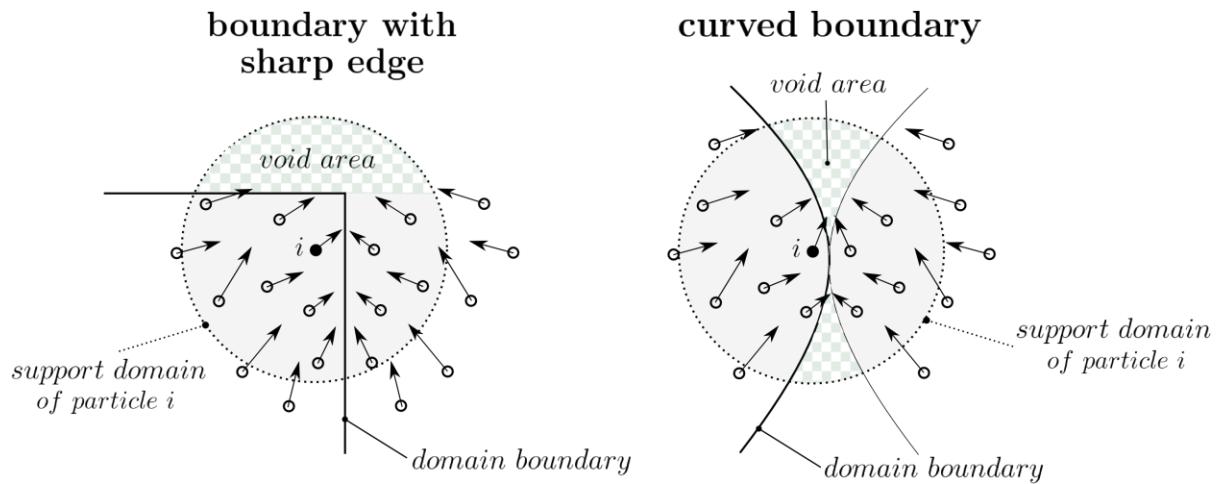


Figure 4.6: The mirror particle boundary formulation does not completely remove particle deficiency if the boundary has sharp edges (left) or is curved (right).

To solve this problem, a combination of mirror particles and repulsive forces was proposed [26] for the treatment of solid boundaries. Particles that create repulsive forces are placed directly on the boundary and mirror particles are created as described above.

4.7.4 First Principal Boundary Conditions

Another way of treating boundary conditions was introduce recently in [42]. It utilizes the surface integral in Eq. (56) that is always assumed to be zero in standard SPH. This assumption of standard SPH is not valid if the support domain of an SPH particle intersects with a boundary. Therefore, in this boundary formulation, the surface integral is evaluated, leading to a consistent way of handling boundaries. For example the SPH version of the continuity equation according to Eq. (64) needs to be extended by the surface term if the latter is non-zero,

$$\frac{d\rho_i}{dt} = \sum_j m_j (v_i - v_j) \nabla_i W_{ij} + \int_{S_b} \rho_b W_{ib} \mathbf{n} \cdot (v_i - v_b) dS_b . \quad (93)$$

Herein, S_b is the boundary surface within the support domain of particle i , ρ_b is the density on the boundary, W_{ib} is the kernel function evaluated at particle – boundary distance and v_b is the boundary velocity. However, this formulation can only be used efficiently if the integral over the boundary surface does not have to be determined at runtime. Solutions can easily be found for a piecewise constant density ρ_b on the boundary. This can be achieved by evaluating the density from known fluid particle properties for a number of points on the boundary and estimating the density in-between using a rectangular approximation fashion. In this way the surface integral can be replaced by a number of N_b simple integrals that can be easily evaluated,

$$\begin{aligned} \frac{d\rho_i}{dt} = & \sum_j m_j (v_i - v_j) \nabla_i W_{ij} . \\ & + \sum_{k=1}^{N_b} \int_{S_{bk}} \rho_{bk} W_{ib} \mathbf{n} \cdot (v_i - v_b) dS_{bk} \end{aligned} \quad (94)$$

The SPH formulations for the conservation of momentum (Eq. (95)) and energy (Eq. (96)) can be constructed similarly.

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) \nabla_i W_{ij} + \mathbf{n} \sum_{k=1}^{N_b} \int_{S_{bk}} \rho_{bk} W_{ib} \left(\frac{p_b}{\rho_b^2} + \frac{p_i}{\rho_i^2} + \Pi_{ib} \right) dS_{bk} \quad (95)$$

$$\frac{d\mathbf{u}_i}{dt} = \frac{1}{2} \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) v_{ij} \nabla_i W_{ij} + \frac{1}{2} \sum_{k=1}^{N_b} \int_{S_{bk}} \rho_{bk} W_{ib} \left(\frac{p_b}{\rho_b^2} + \frac{p_i}{\rho_i^2} + \Pi_{ib} \right) \mathbf{n} \cdot \mathbf{v}_{ib} dS_{bk} \quad (96)$$

All field variables like density, pressure and internal energy, which are used in the surface integrals over the boundary, can be determined for each point on the boundary using a weighted summation over neighboring particles according to Eq.(97),

$$f_b = \frac{\sum_j m_j f_j W_{bj}}{\sum_j m_j W_{bj}}. \quad (97)$$

Here, f is the field variable and the summation is over all particles within a distance of κh . It is reported that this first principal boundary conditions remove particle deficiency induced instabilities, leading to accurate simulation results. This formulation was not used in the simulations because of the shortcomings that are explained in section 4.7.5.

4.7.5 Simplified First Principle Boundary Conditions

The first principal boundary conditions described above have the disadvantage that the principle of the limited support domain is violated. For example two particles having a distance of more than κh can still interact with each other over the boundary surface integral as illustrated in Figure 4.7.

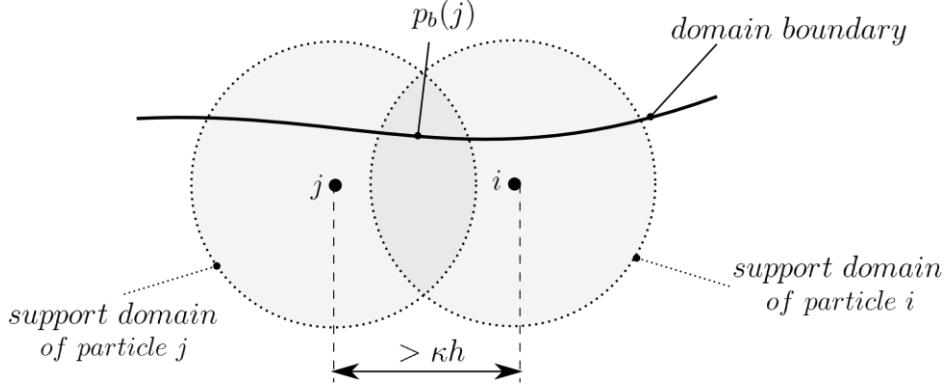


Figure 4.7: Interaction of particles j and i over field variable values on the boundary. The pressure, p_b , at one point of the boundary depends on properties of particle j . Since the point on the boundary is within the support domain of particle i , the particles interact indirectly.

The pressure on the boundary point depends on the pressure of particle j and the motion of particle i depends on the boundary pressure. In this way the two particles interact with each other even though the distance between them is greater than κ times the smoothing length.

Besides this rather theoretical issue, the first principal boundary conditions have another major disadvantage. They cannot completely inhibit boundary penetration especially for high pressure gradients or velocities near a boundary. If the pressure gradients are high, the pressure, p_b , determined according to Eq. (97), can be much smaller than the pressure in the domain. This leads to high accelerations towards the boundary can result in boundary penetration. For this reason, additional repulsive forces need to be applied to ensure that no particle penetrates a solid boundary. One way to do so is to simply multiply the theoretical pressure, p_b , on the boundary by a factor greater than one. On the other hand if a crude assumption like this is made, the surface integration in Eq. (94) does not have to be very exact. Therefore, the new simplified first principal boundary conditions assume the pressure on the boundary to be constant for each particle and equal to the pressure associated with the particle itself. The intuitive interpretation is that the particle, as it approaches a solid wall,

more and more interacts with itself similar to an elastic ball bouncing on the floor. In order to prevent boundary penetration, the surface pressure, p_b , is chosen slightly higher than the particle pressure, p_i . This automatically causes repulsive forces normal to the boundary which result from the surface integral in Eq. (95) and determines the particles to stay inside the domain. For hydrodynamic test simulations a boundary pressure of, $p_b = p_i + kp_0$, for $k = 0.1$, was used. Herein, p_0 is the atmospheric pressure. Preventing boundary penetration successfully and leading to stable simulation results, this formulation is later used in all the liquid simulations of section 5.2.3. However, the factor k might have to be chosen higher or lower to meet the specific requirements of different simulations. Using the assumption above Eqns. (95), (96) and (97) can be simplified to the following three expressions for the conservation laws,

$$\begin{aligned} \frac{d\rho_i}{dt} &= \sum_j m_j v_{ij} \nabla_i W_{ij} \\ &\quad + \mathbf{n} \cdot v_{ib} \rho_i \int_{S_b} W_{ib} dS_b \end{aligned} \tag{98}$$

$$\begin{aligned} \frac{dv_i}{dt} &= - \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) \nabla_i W_{ij} \\ &\quad + \mathbf{n} \left(2 \frac{p_b}{\rho_i} + \rho_i \Pi_{ib} \right) \int_{S_b} W_{ib} dS_b \end{aligned} \tag{99}$$

$$\begin{aligned} \frac{du_i}{dt} &= \frac{1}{2} \sum_j m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + \Pi_{ij} \right) v_{ij} \nabla_i W_{ij} \\ &\quad + \mathbf{n} \cdot v_{ib} \left(\frac{p_b}{\rho_i} + \frac{\rho_i}{2} \Pi_{ib} \right) \int_{S_b} W_{ib} dS_b \end{aligned} \tag{100}$$

It can be seen that due to the assumption made, the surface integrals in the preceding equations depend only on the kernel function and the boundary surface. Another approximation made is that the surface integral is evaluated over a straight line in two, and a

plain surface in three dimensions. This requires the boundary to have small curvature and to be smooth at least over a distance of the smoothing length, h . In this way, surface integrals can be calculated numerically and approximated by a function which later can be used in the simulation. This approximation step will be discussed below for the cubic kernel function.

According to Eq. (55) and the divergence theorem, the surface integral in Eqs. (98), (99) and (100) can simply be replaced by the integrand if the problem domain is one-dimensional. Therefore no additional work needs to be done here as the kernel can be evaluated efficiently at runtime. For a two-dimensional problem the surface integral reduces to a one-dimensional integral that needs to be evaluated. However, it is reasonable to determine a general solution that can be used to replace the time-expensive integration in the actual simulation. This integration was done numerically using Matlab since no analytical solution could be found. The surface integral is necessary if the particles is boundary-particle distance, R_b , is smaller than κh and the support domain gets truncated by the boundary. If the kernel is two-dimensional, this means that the integral is equal to the surface area, visualized in Figure 4.8.

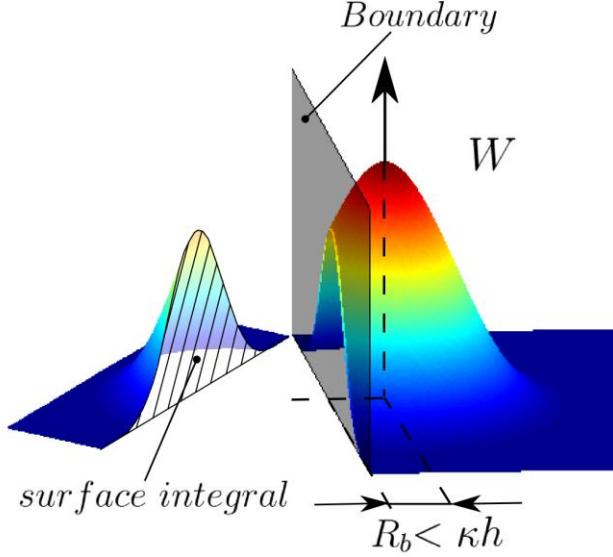


Figure 4.8: Surface integral over the two-dimensional SPH kernel, W , on the domain boundary.

The integral can be expressed as follows,

$$\int_{-\kappa h}^{\kappa h} W_{ib} dx_b \quad \left| \begin{array}{l} W_{ib} = W(\mathbf{x}_i - \mathbf{x}_b, h) \\ \mathbf{x}_i = [0, R_b]'; \quad \mathbf{x}_b = [x_b, 0]', \end{array} \right. \quad (101)$$

whereas the numerical solution is plotted over a set of 300 different particle-boundary distances, R_b , as shown in Figure 4.9. Herein, a smoothing length of $h = 1$ and a kernel parameter of $\kappa = 2$ is used.

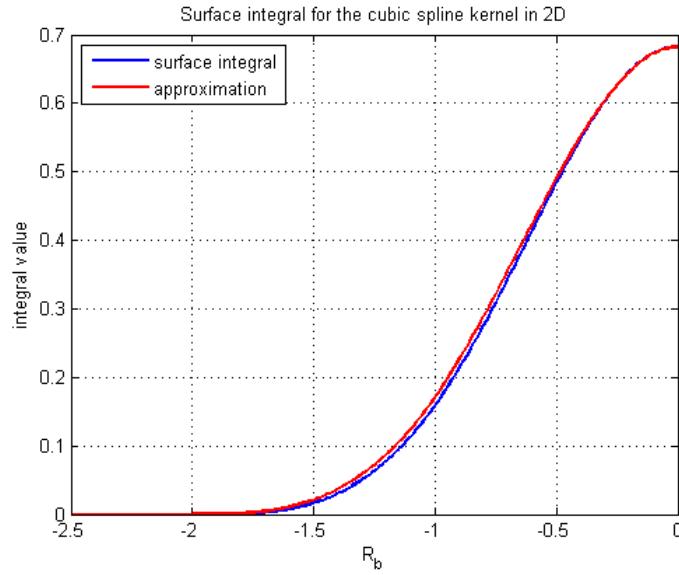


Figure 4.9: Surface integral over the normalized two-dimensional cubic spline kernel and the approximation function.

It can also be seen that the result can be approximated through the following expression, which is general for all parameters.

$$\int_{S_b} W_{ib} dS_b = \int_{-\kappa h}^{\kappa h} W_{ib}^{2D} dx_b \approx \frac{W_{ib}^{2D}}{W_0^{1D}} \quad (102)$$

Herein, W_0^{1D} is the peak value of the one-dimensional kernel and W_{ib}^{2D} is the two-dimensional kernel evaluated at particle-boundary distance. This approximation can then be used to replace the surface integral for two-dimensional problems since it can be evaluated very quickly at runtime.

In three dimensions, it is harder to visualize the surface integral but it can be expressed similarly as in the case of two dimensions.

$$\int_{-\kappa h}^{\kappa h} \int_{-\kappa h}^{\kappa h} W_{ib} dx_b dy_b \quad \left| \begin{array}{l} W_{ib} = W(\mathbf{x}_i - \mathbf{x}_b, h) \\ \mathbf{x}_i = [0, 0, R_b]'; \quad \mathbf{x}_b = [x_b, y_b, 0]' \end{array} \right. \quad (103)$$

Here, the surface integral is two-dimensional and needs to be evaluated numerically as well. Figure 4.10 shows the results plotted over a set of 300 different particle-boundary distances, R_b .

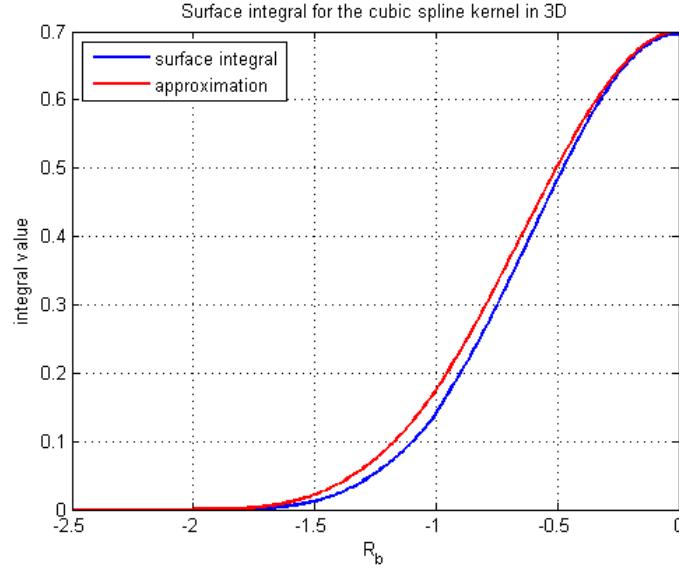


Figure 4.10: Surface integral over the normalized three-dimensional cubic spline kernel and the approximation function.

An approximation for this curve can be found similarly as in two dimensions,

$$\int_{S_b} W_{ib} dS_b = \int_{-\kappa h}^{\kappa h} \int_{-\kappa h}^{\kappa h} W_{ib}^{3D} dx_b dy_b \approx \frac{W_{ib}^{3D}}{W_0^{2D}}, \quad (104)$$

where W_0^{2D} is the peak value of the two-dimensional kernel and W_{ib}^{3D} is the three-dimensional kernel evaluated at particle-boundary distance. Replacing the surface integral through this approximation leads to fast boundary treatment.

4.8 Fluid-Structure Interaction

A very attractive feature of SPH is its intuitive handling of fluid-structure interactions. Interaction forces can be obtained either directly (in the case of repulsive boundary forces),

or indirectly by multiplying boundary induced particle accelerations with the particle mass (in case of the other boundary formulations). According to Newton's third law, the force on the boundary or structural body is equal and opposite relative to the force on the particle. A summation over all particle-structure forces associated with one part leads to the absolute force on it. The rotational dynamics of a body can be expressed in a simple way if torques are given with respect to the body's center of mass. Therefore, torques associated with each particle-structure force are calculated using the following relation between force, \mathbf{f}_i , and torque, τ ,

$$\tau_i = (\mathbf{x}_{bi} - \mathbf{x}_{CM}) \times \mathbf{f}_i . \quad (105)$$

Herein \mathbf{x}_{bi} is the point at which the fluid-structure force associated to one particle acts, and \mathbf{x}_{CM} is the body's center of mass position. Summation over all these particle-structure torques leads to the total torque on the body.

Although, both deformable as well as rigid bodies can be considered for fluid-structure interactions, for reasons of simplicity, only rigid bodies will be discussed here. Hence, the following kinetic relations can be used to describe the dynamics of rigid bodies under the described fluid dynamic forces [43].

Translation of the center of mass:

$$\mathbf{a}_{CM}(t) = \frac{\sum \mathbf{f}_i(t)}{M} \quad (106)$$

$$\mathbf{v}_{CM}(t) = \mathbf{v}_0 + \int_0^t \mathbf{a}_{CM}(t) dt \quad (107)$$

Rotation:

$$\mathbf{I}(t) \cdot \dot{\boldsymbol{\omega}}(t) = \sum \boldsymbol{\tau}_i(t) \quad (108)$$

$$\boldsymbol{\omega}(t) = \boldsymbol{\omega}_0 + \int_0^t \dot{\boldsymbol{\omega}}(t) dt \quad (109)$$

Hereby, the translational acceleration is determined according to Eq. (106) from the summed particle-structure forces, f_i , and the mass, M , of the rigid body. The center of mass acceleration, a_{CM} , needs to be integrated to obtain the velocity, v_{CM} . The rotational speed of the body can also be determined over time integration whereas the rotational acceleration, $\dot{\omega}$, is calculated according to Eq. (108) using the body's moment of inertia tensor, $I(t)$, and the summed torques, τ_i . Time integration can be done using explicit Euler integration or other more sophisticated methods for higher precision and stability requirements. The position of every point on the rigid body can then be updated for example in the following manner,

$$\mathbf{x}_{bi}(t) = \mathbf{x}_{bi}(t - \Delta t) + \mathbf{v}_{CM}(t)\Delta t + \mathbf{S}(\boldsymbol{\omega}\Delta t) \cdot \mathbf{r}_{bi}, \quad (110)$$

which is based on explicit Euler integration. $\mathbf{S}(\boldsymbol{\omega}\Delta t)$ is the Cartesian rotation matrix and \mathbf{r}_{bi} is the vector from the center of mass to the point on the body. This formulation can only be used if $\boldsymbol{\omega}\Delta t$ is very small, which is true in the conducted simulations. Otherwise different representations like for example quaternions might be more adequate. The moment of inertia tensor, $I(t)$, in Eq. (108) also needs to be updated at each time step since it changes as the body rotates.

$$\mathbf{I}(t) = \mathbf{S}(\boldsymbol{\omega}\Delta t) \cdot \mathbf{I}(t - \Delta t) \cdot \mathbf{S}(\boldsymbol{\omega}\Delta t)^T \quad (111)$$

The velocity \mathbf{v}_{bi} of every point on the body necessary to determine the body-fluid interaction force can be calculated according to Eq. (112).

$$\mathbf{v}_{bi}(t) = \mathbf{v}_{CM}(t) + \boldsymbol{\omega}(t) \times \mathbf{r}_{bi} \quad (112)$$

The geometry of simple structural parts can be described analytically which makes the search for particle-structure interactions straight forward. However, given that most bodies usually have more complicated geometries, it would be desirable to be able to load mesh data from

CAD geometries. It should be noted that both the boundary's position and its normal vector have to be known in order to use the boundary formulations described above. This information can be obtained directly from a typical triangular surface tessellation. Yet, it is not optimal to use mesh data directly in the simulation since a separate particle-triangle collision detection would have to be implemented. Hence a method called spherical decomposition is used to replace CAD triangular meshes by particles for which the SPH particle collision detection doesn't need to be vastly modified [44]. Particles are placed in the center of each surface mesh triangle and surface normal vectors are saved together with particle positions. The force's position, \mathbf{x}_{bi} , can be calculate according to Eq. (113) from the SPH particle position, \mathbf{x}_i , and the surface normal, \mathbf{n} , associated with the closest boundary particle.

$$\mathbf{x}_{bi} = \mathbf{x}_i - \mathbf{n}((\mathbf{x}_i - \mathbf{x}_{bi}) \cdot \mathbf{n}) \quad (113)$$

In this way, particle-structure forces can be calculated for the boundary particle that is closest to the respective SPH particle.

4.9 ***SPH for Acoustic Simulations***

SPH has attractive features useful not only in fluid dynamics but also in acoustics. An intensive literature research showed that SPH has only once been considered for acoustic simulations [45], where it was used for the calculation of room reverberation. Details about this work will be discussed in section 6. However, SPH is not limited to architectural acoustics, its strengths of might be especially beneficial for more complex acoustic problems.

Due to its meshless nature, SPH can handle complex domain topologies without the typical problems of grid methods for which constructing the grid for a complex geometry is

difficult. This feature makes SPH attractive for acoustic simulations where the boundaries are often quite complex. Examples can be found in room acoustics with complex furniture surfaces, or in music instruments where complex surfaces dominate the sound. Furthermore, the resolution of SPH simulations can easily be adjusted by using more particles in the regions of interest.

Determining the location of deformable boundaries and moving interfaces is difficult in a fixed Eulerian grid used in Finite Differences. FEM doesn't have these problems but large deformations require re-meshing techniques which are time consuming and error prone. In meshless Lagrangian methods, like SPH, the problem domain moves naturally with the flow and there is no predefined mesh to provide connectivity between particles. This is closer to the real physics and makes SPH ideal for changing domain topologies and large deformations. In acoustics, there is a whole branch of problems where these properties of SPH can be useful. One area is the field of internal combustion engines where the problem domain, which could be the combustion chamber, undergoes large deformations due to the piston motion, with changing domain topology, caused by valve motion [3].

SPH particles carry material properties and are free to move due to external forces and internal interactions, which makes it easy to simulate multiphase flows with moving material interfaces. It is interesting to note that the speed of sound, c , is nowhere used in the whole SPH formulation. On the contrary it evolves naturally from the fluid properties given at each particle. Therefore SPH is perfectly suited to simulate sound propagation through gases with a temperature gradient resulting in a gradient in the speed of sound. It can also be used to simulate the propagation of sound through multiphase flows of fluids or gases with different

sound speeds. This leads back to acoustic simulations of internal combustion engines where not only different gases but also high temperature gradients in the gases affect the sound propagation. Another application is the refraction of sound waves in the atmosphere due to temperature gradients [46].

Unlike BEM, FEM or FDTD, which in acoustics are typically used to solve the wave equation, SPH can resolve the whole flow process and therefore account for bulk flow, viscosity effects and aero-acoustic phenomena in general. Since standard SPH for hydrodynamics is based on the same conservation laws as the Navier-Stokes equations, it can be considered as a simulation tool for a wide variety of fluid dynamic problems including DNS and Hybrid simulations. Additionally there have been studies about the accuracy of SPH simulations involving turbulent flows which are a main cause of flow induced sound generation. These studies came to the conclusion that SPH is capable of simulating turbulent effects but the standard algorithm has to be enhanced in order to obtain satisfactory results [47, 48]. Sound generation due to turbulent flows is an important field since it is one of the major sources of noise in cars, trains and airplanes. The use of SPH in DNS or Hybrid simulations for those kinds of problems could prove beneficial in special cases even though obtaining a solution will likely require longer simulation times compared to current DNS methods. The same resolution as in other DNS methods is necessary, which leads to high particle numbers and computation times. Nonetheless the other attractive attributes of the meshless and Lagrangian nature of SPH are still maintained.

It was shown in section 2.1 that the linear wave equation solved by the majority of current simulation methods is based on a number of assumptions such as low pressure

amplitudes and gradients. This limits the range of applications since high pressure amplitudes and incontinuities, as given in shock waves, result in poor results. However, SPH is based on conservation laws and constitutive relations that don't have to be linearized. Thus it should be possible to model nonlinear effects accurately. As explained in section 3.4.3 the only limitation for SPH in the field of nonlinear acoustic is dispersion. The cause for acoustic dispersion cannot be modeled by the conservation laws and therefore for problems which involve high gradients have to be tressed with caution. The formalism of SPH however provides a numerical dispersion which can be adjusted over parameters such as smoothing length or artificial viscosity. For this reason the use of carefully tuned SPH parameters can lead to good results for shock waves. This was shown for explosions [49] as well as shock and impact flows [50].

Chapter 5

Implementation of SPH

The described SPH formalism for fluids is implemented in two different ways. The first is a serial implementation in Matlab and the second is a parallel implementation leveraging the computing power of Graphics Processing Units (GPUs). The implementation uses C++ code and NVIDIA's Compute Unified Device Architecture (CUDA) application programming interface (API). Details about each implementation and the programming environments can be found in sections 5.1 and 5.2.

Common to all SPH implementations is a search algorithm, that at each time step finds the interaction partners of each SPH particle. This step, called nearest neighbor search, is necessary because the derivative of a field function is determined based on the field function values of neighboring particles as shown in section 4.1. A simple brute force search through all particles would result in a very inefficient $O(N^2)$ -algorithm, N being the number of bodies. Thus, this problem needs to be addressed with a different search algorithm that scales better. Along with other list or binary tree based algorithms, a method known as spatial subdivision search has lately gained popularity since it is highly parallelizable. In this method, the volume occupied by SPH and boundary particles is divided into a grid of smaller bins. These bins have a side length of κh (see Figure 5.1), following Monaghan and Latanzio [38]. Herein, κ is equal to 2 for the cubic spline kernel.

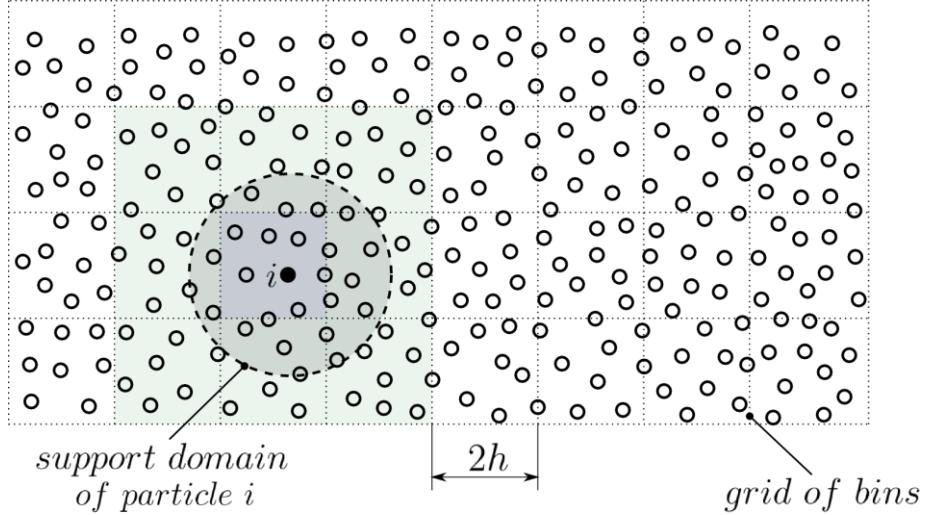


Figure 5.1: Binning step of spatial subdivision. Since a cubic spline kernel is used, the bin side length has to be equal to $2h$ in order to obtain a fast algorithm along with capturing every possible particle interaction.

This has the advantage that for a particle located inside one bin, interaction partners can be found only in directly neighboring bins. Thus for each SPH particle the particles contained in 9 or $27sh$, bins have to be checked for potential interaction in two and three dimensions respectively. Given the small number of bodies per bin, the brute force search can then be performed quickly. In this way the number of calculations per simulation step and therefore the computational time diminishes significantly from $O(N^2)$ to $O(N \log(N))$.

More recently a new extended spatial subdivision search algorithm has been implemented for contact mechanic simulations. This extended method is reported to scale linearly [51], and has the additional advantage that bodies of different size can be checked for interaction very efficiently. For SPH simulation this means that varying smoothing lengths can be handled without leading to greatly increased computation times as given in the basic method [38]. The extended method is implemented for two dimensional problems in section 5.1. Only the basic method was used for three dimensional simulations.

Once interaction partners have been found for each SPH particle, field function derivatives can be calculated using the SPH specific kernel function approach as described in section 4.1. As seen in section 4.7, the effect of boundaries or fluid-structure interaction needs to be addressed in the same step since it also affects field function derivatives of particles near interfaces. As soon as the derivatives are determined the next step of the simulation can follow where integration, as described in section 4.4, is used to resolve the time evolution of the simulated system. Afterwards, the next simulation loop can start with a new nearest neighbor search based on updated particle positions.

Before a simulation can it first has to be set up. This includes particle placement as well as the positioning of fixed or moving boundaries and bodies. Additionally, initial field variables have to be chosen for each SPH particle and the initial velocity of bodies or moving boundaries need to be given. Typically, simulation data needs to be analyzed using other computer programs. Consequently, a user specified subset of simulation data has to be saved while a simulation is running. For hydrodynamic simulations it is often enough to save only position information which later can be visualized in a rendering program. For acoustic simulations pressure and velocity data is usually most important and needs to be saved.

The functionality of basic SPH implementations with all simulation stages is illustrated in Figure 5.2.

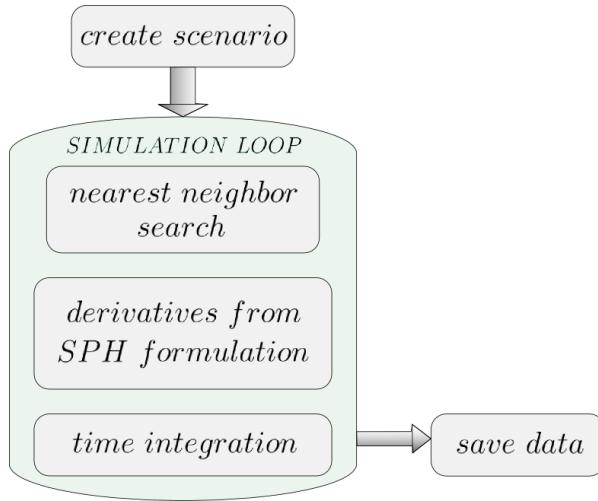


Figure 5.2: Basic SPH formulation. After the particles have been created their time evolution is simulated in the simulation loop while data can be saved for further analysis.

5.1 One and Two-Dimensional Implementation

Due to its simplicity and flexibility the numerical computing environment Matlab was chosen for a first rapid assessment of SPH. Matlab provides several visualization tools that can be included in the time stepping scheme to monitor field variables while the simulation is running. The framework outlined in Figure 5.2 has been implemented in m-files, with the exception of the fluid-structure interaction. Rigid body motion cannot be simulated and boundaries are limited to simple geometries. Nevertheless, the program has all the functionality necessary to conduct basic numerical experiments to gauge information about the applicability of SPH for acoustic simulations. The structure of the program is summarized in the following flow chart, which shows all m-files and their interdependency.

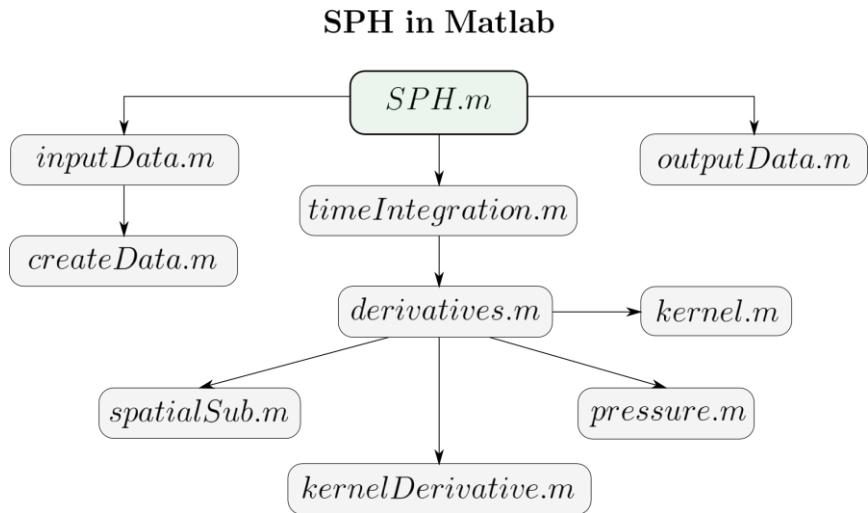


Figure 5.3: Hierarchic structure of the Matlab m-files of the SPH program.

The program uses the built-in Matlab sorting function “sortrows.m” in the nearest neighbor search step and some visualization tools along with the standard algebraic functions. Benefiting from the fact that Matlab already takes care of all memory allocation and other administrative work, the code is relatively short and has about 600 lines. Given that all computational steps are processed sequentially the performance of the Matlab code is relatively poor. Consequently, it is reasonable to simulate only systems with a relatively small number of particles. The time step length for a systems in the order of 10,000 SPH particles is about 30 seconds making simulations with higher discretization inefficient. As a result the Matlab code is limited to one and two-dimensional SPH simulations since three dimensional simulations with reasonable resolution are necessarily coupled to a much higher number of particles. The performance of Matlab programs can be increased by the use of so called Matlab extension (MEX) files that include C-code or CUDA GPU code. These MEX files can be used to speed up the computationally intensive parts of Matlab programs. In the case of the SPH program almost all steps are computationally expensive meaning that MEX-

files would have to be generated at least for: nearest neighbor search, the evaluation of field function derivatives and the time integration. Conversely, if the computation has to be done on the GPU using the CUDA API, this will lead to a high amount of data transfer between CPU and GPU. This scenario is not optimal in terms of program efficiency and therefore no attempt was made to optimize the Matlab code. Instead a C++ and CUDA API based program was developed as reported in section 5.2.

5.1.1 2-D Hydrodynamic Test Simulation

Several hydrodynamic test simulations have been conducted to test all parts of the program. Due to the extensive use of SPH in hydrodynamics a lot of reference data is available to validate results. Another advantage is the human intuition about fluid motion. Although these insights can be used to decide if simulation results are reasonable or not and turned out to be especially helpful in the debugging process, they cannot replace systematic validation.

A classical SPH test case, the 2-D dam break experiment, is run in order to test the program. A number of 7,225 fluid particles are placed in a square at the left side of a box. The particles are arranged on a rectangular grid with spacing d and a Gaussian random component with a standard deviation of 0.1 d in both vertical and horizontal direction. The walls of the box are modeled with dynamic boundary particles (see section 4.7.1.), since it is straight forward and requirements on the boundary conditions are relatively low in this experiment. The main purpose of the walls is only to prevent boundary penetration. The motion of the particles under gravity is simulated over time and the results are qualitatively compared to results in literature [52]. Figure 5.4 shows the time evolution of the particle

positions and the associated pressure, the latter visualized by different colors from low pressure, blue, to high pressure, red. The following table shows the parameters and geometry of the dam break experiment.

Table 5.1: Parameters and geometry for the dam break experiment.

Artificial viscosity parameters [29]:		
$\alpha = 0.001$	$\beta = 0$	$\gamma = 0.01$
equation of state: $p = B \left(\left(\rho / \rho_0 \right)^n - 1 \right)$		
$B = 3000$	$n = 7$	$\rho_0 = 1000 \text{ kg/m}^3$
smoothing length:		2.5 mm
smoothing kernel:		cubic spline
time step size:		$5.0e^{-5} \text{ s}$
number of particles:		7,225
particle mass:		$3.52e^{-3} \text{ kg}$
average particle spacing:		1.875 mm
standard dev. of Gaussian distribution:		0.1875 mm
width and height of the fluid square:		0.153 m
distance between the two walls:		0.425 m
total simulation time, T_{end} :		0.25 s

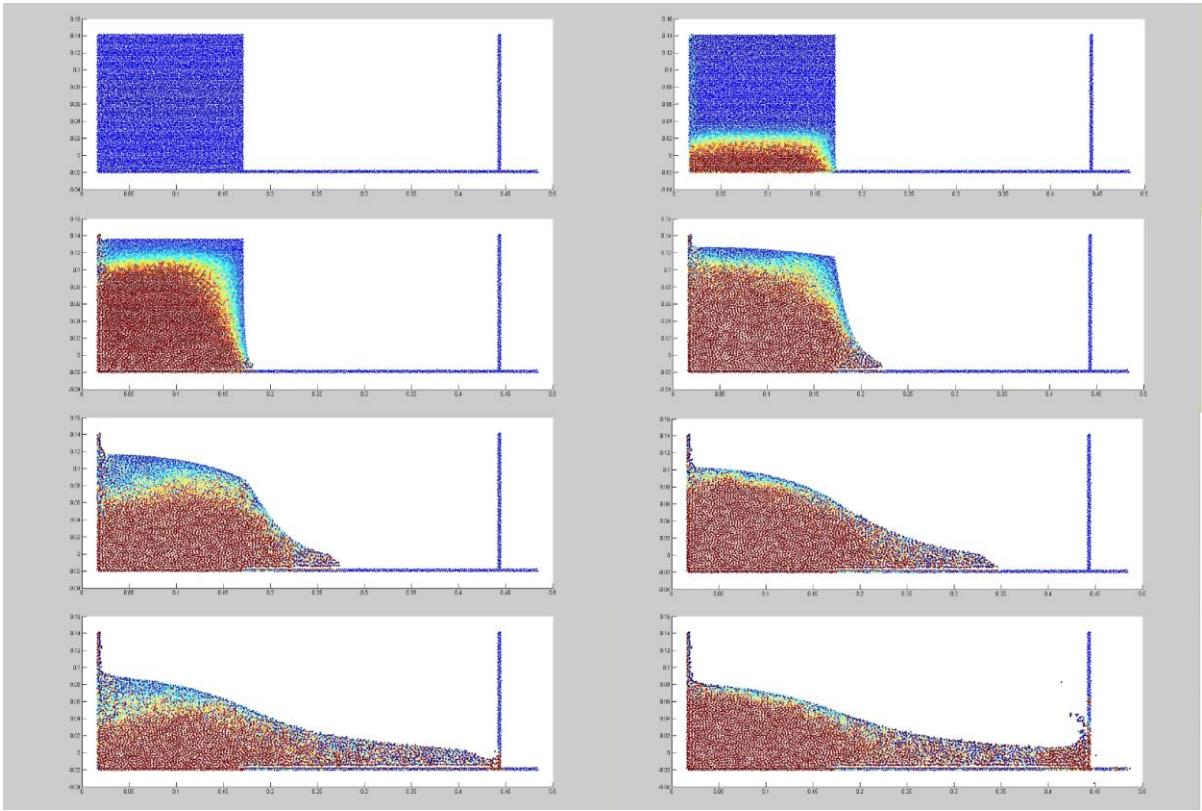


Figure 5.4: Pressure wave propagation and free surface flow due to gravity in the dam break experiment.

During the first part of the simulation a pressure wave runs through the water starting at the ground and goes upwards. The pressure gradient generates motion in the particles. The pressure wave in the simulation is not realistically fast because the equation of state is chosen to be more compressible than for real water. This is necessary for stability to allow reasonably large step size. After the pressure gradient has set the particles in motion, a flow pattern can be observed that is very close to that of water.

5.2 Three-Dimensional Implementation

Three-dimensional simulations with good resolution are typically connected to discretizations of over one million SPH particles. The fact that the above dam break

simulation with less than 10,000 SPH particles took approximately 25 hours on a modern desktop PC clearly shows the need of a faster implementation.

Clearly, it is advantageous to use another programming language like FORTRAN or C++ but the biggest potential lies in the high level of parallelism associated with the method. As previously mentioned, the Matlab implementation performs all steps for each particle sequentially; however, all computationally intensive parts of the SPH algorithm; i.e. nearest neighbor search, evaluation of derivatives and time integration, can be processed in parallel. This means that many computations can be performed at the same time by many different processing threads. Thus parallel computing has the potential to dramatically increase computational efficiency, allowing three-dimensional simulations with a large number of particles to be run in a reasonable amount of time.

A very cost effective way to leverage parallel computing is the use of graphics processing units (GPUs). These GPUs, manufactured in large numbers for use in personal computers, typically fulfill graphics purposes. However, ever since GPU-programming tools became available, GPUs have become increasing popular in the field of high performance computing. GPU computing was successfully leveraged to increase computational speed in a variety of fields. In particular multibody dynamics with frictional contacts employs computational steps similar to those used in SPH. The nearest neighbor search, being the first essential step in SPH, is very similar to the collision detection performed in multibody dynamic simulations. The second step in multibody dynamics is the calculation of collision and friction forces where the SPH equivalent would be the evaluation of the momentum equation. Finally, in both cases, integration has to be used to determine the time evolution of

the system. A detailed discussion of the related topic of multibody dynamics with frictional contacts can be found in [51] and [44].

5.2.1 GPU-Programming

Due to its original use in computer graphics the structure of the GPU is such that many arithmetic instructions can be performed relatively slow but simultaneously. At the other end of the spectrum, the central processing unit (CPU) can sequentially perform complex computing tasks very fast. A measure of computational performance is the rate at which Floating-point Operations can be performed (FLOP rate). Figure 5.5 shows the evolution of the peak FLOP rate over the past six years for both GPU and CPU.

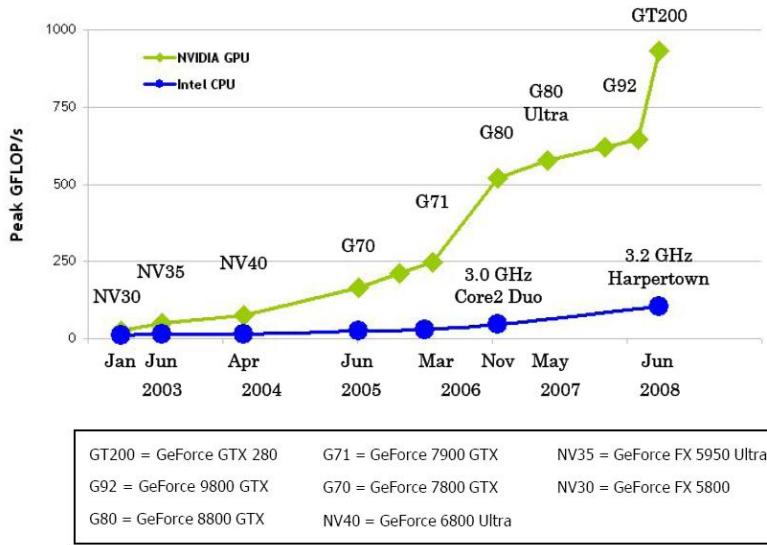


Figure 5.5: Speed comparison between GPUs and CPUs in terms of FLOP rate [53]. While for the CPU the Flop rate is given for double precision arithmetic, the values for the GPU have been evaluated for only single precision computation. However, the slope of the GPU FLOP rate is significantly higher.

This demonstrates that current GPUs can perform calculations much more quickly than CPUs [53]. However, the majority of the current GPU generation can only perform single

precision calculations while CPUs typically work in double precision. This can lead to problems as described in section 6.3.

In this section, the GPU hardware discussion concentrates on the NVIDIA TESLA C1060 model. This card has one stream processor array (SPA) which is composed of ten texture processor clusters (TPC). Each TPC has three stream multiprocessors (SM), each of which has eight scalar processors (SP). In total, the TESLA card has 240 scalar processors [54]. See Figure 5.6 for a diagram of the GPU hardware.

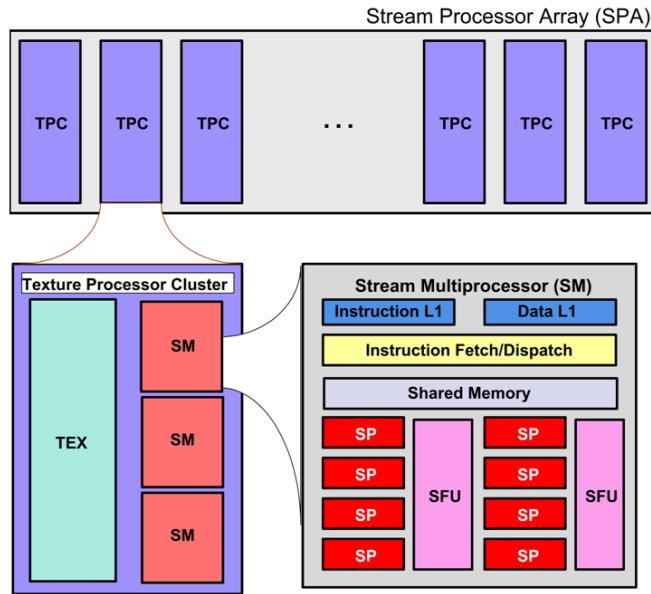


Figure 5.6: Hardware structure of GPUs [53].

The computational paradigm of the GPU is called single-instruction multiple data (SIMD) meaning that the GPU performs the same computation on multiple data elements simultaneously. In GPU programming the use of memory has a crucial influence on performance since there are several types of memory, each with its own inherent characteristics such as latency and bandwidth. Figure 5.7 illustrates the memory structure of a GPU.

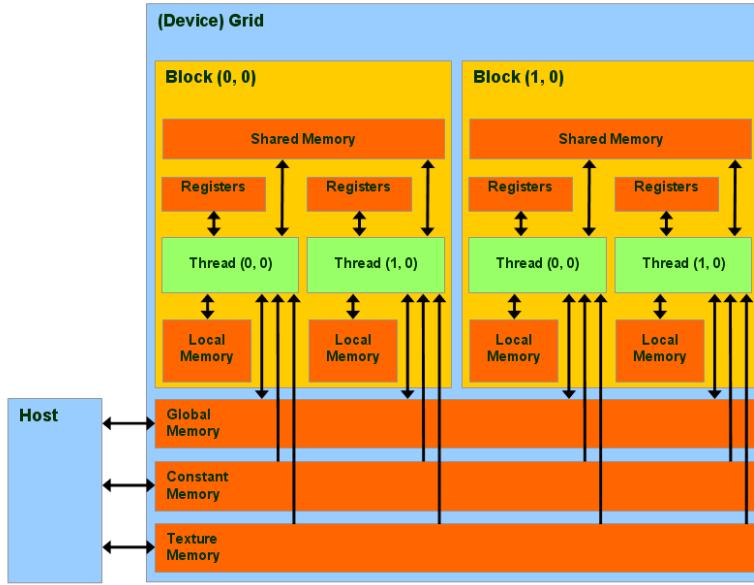


Figure 5.7: Memory structure of GPUs [53].

- Registers (read/write per thread): Used to hold temporary data while performing computations. Each computational thread has its own registers.
- Shared memory (read/write per block): Used to hold data which needs to be shared by the threads grouped together in a block.
- Global memory (read/write per grid): Used to hold input and output data, accessible by all threads.
- Constant memory (read only per grid): Used to hold constants, accessible by all threads.
- Texture memory (read only per grid): Used to hold data arranged in a 2D grid, accessible by all threads.

Access speeds vary significantly with memory type explaining why memory use plays a central role in GPU programming. Global memory is generally the conduit that is used for CPU/GPU memory transfers. Given the limited host-GPU connection speed, the time required to transfer data to and from the GPU can be significant. Thus the GPU is most efficient when the ratio of computations per piece of data is high.

In the work presented here, the GPU was programmed using NVIDIA's Compute Unified Device Architecture (CUDA). CUDA code is organized into functions, or kernels, which are launched by the host as a grid of blocks. Each block consists of threads that can cooperate with each other through shared memory [53]. Figure 5.8 demonstrates the computational outline of a CUDA program.

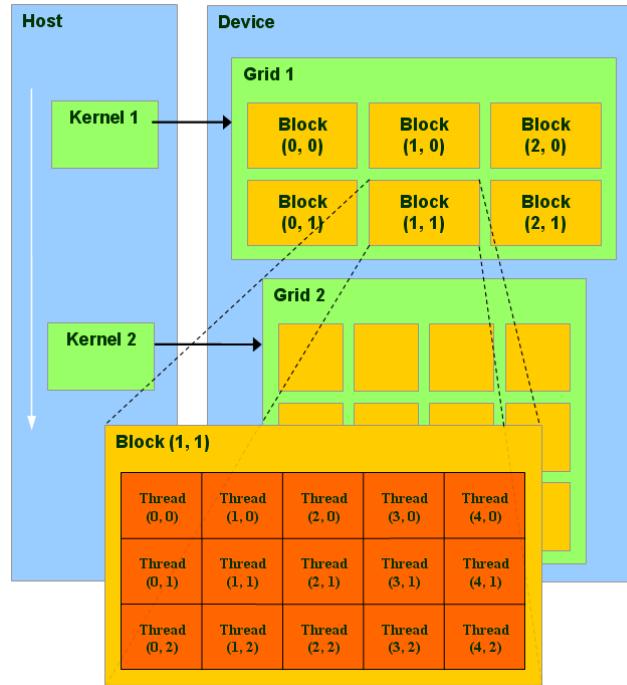


Figure 5.8: Structure of a CUDA program [53].

Parallelism is exploited on a thread basis because each thread executes the same instructions as defined in the kernel. While conditionals can be used to cause different threads to perform different computations, their use will adversely affect performance.

5.2.2 Program Structure

As already mentioned, the SPH algorithm for three dimensional problems is programmed using the CUDA API which can be used to execute C code on the GPU. Additionally, NVIDIA provides a collection of tools and examples in its CUDA software developing kit (SDK). One demo in the SDK called “particles” is a program for multibody dynamics with frictional contacts. This program uses similar steps as required in an SPH implementation so it became the starting point of the three dimensional SPH program.

The file structure and the parallel collision detection of the particle demo are recycled with only minor adaptations. One of them is to set the side length of the bins equal to $2h$ where h is the longest smoothing length of all SPH particles in the system. Another nice feature of the particle demo is that the position of particles is displayed while simultaneously being updated with the simulation. This visualization, which uses Open GL, is augmented in order to display the pressure equivalent color of each particle. Kernels for the evaluation of field function derivatives according to Eqs. (64), (69) and (77) and leap frog time integration, as demonstrated in section 4.4, are implemented. These steps, which form the basic SPH functionality, are expanded through other functions. A function for the loading of boundary particles allows creating complex boundary geometries or movable rigid bodies. The files that contain all data associated with these boundary particles are created using a Matlab program developed to replace complex shaped bodies in multibody dynamics simulations

through a collection of spheres [44]. The defining triangular mesh can be created in Cubit from SolidWorks models. The program was slightly modified in order to save the surface normal together with the position of each particle. The formulation for rigid body dynamics, explained in section 4.8, is also implemented in order to simulate fluid-structure interaction.

Field variables associated to each SPH particles are saved in large arrays of length N , where N is the number of particles. Accelerations, velocities and positions are saved as float4-variables which contain the values in all three space dimensions plus an unused fourth term. The use of float4 leads to a speed advantage over float3 because reading and writing is more coalesced as explained in [53]. The other field variables, density and internal energy, are saved in float-arrays and another variable which defines the particle type is saved using an integer-array. Particles used to discretize different fluids have distinct type-values prompting the program to treat particles according to predefined rules. Since these arrays are by far the biggest data structures in the SPH program and need to be read and modified frequently it is desirable to use memory with fast access times. While all arrays are stored in the slow global memory, texture memory fetching can be utilized to achieve reduced access times. To this point, however, no further attempts regarding optimized memory use have been made. Leveraging the different memory types of the GPU will presumably lead to highly increased computational efficiency. The reason why this SPH implementation still shows good performance is traced to the fact that the time consuming communication between host and GPU is limited to simulation data that is saved on the hard disc. These data files can be used either in post processing visualizations or further simulation analysis.

The GPU used for most of the simulations is a NVIDIA GeForce 8800 GTX with 128 stream processors working at a clock rate of 600 MHz and using a total of 768 MB of memory. Due to this memory limitation, it is possible to run simulations with up to 3.5 million SPH particles. The implementation on the GPU leads to a relative speed up of 3,500 in comparison to the Matlab program. Without counting external libraries the whole program consists of over 4,000 lines of code, of which about 1,000 lines stem from the CUDA SDK particle demo.

5.2.3 3-D Hydrodynamic Test Simulations

In order to debug and test all parts of the program different hydrodynamic test simulations were created. They all use the new simplified first principle boundary conditions as explained in section 4.7.5. The position of particles determined by the SPH simulation is saved in a text file at different time steps. This information is then used to create a visualization using the ray tracing program Povray [55].

The first example shows nine rendered snapshots of a simulation with 260,000 particles in which a spherical droplet falls into a basin of liquid with water-like properties.

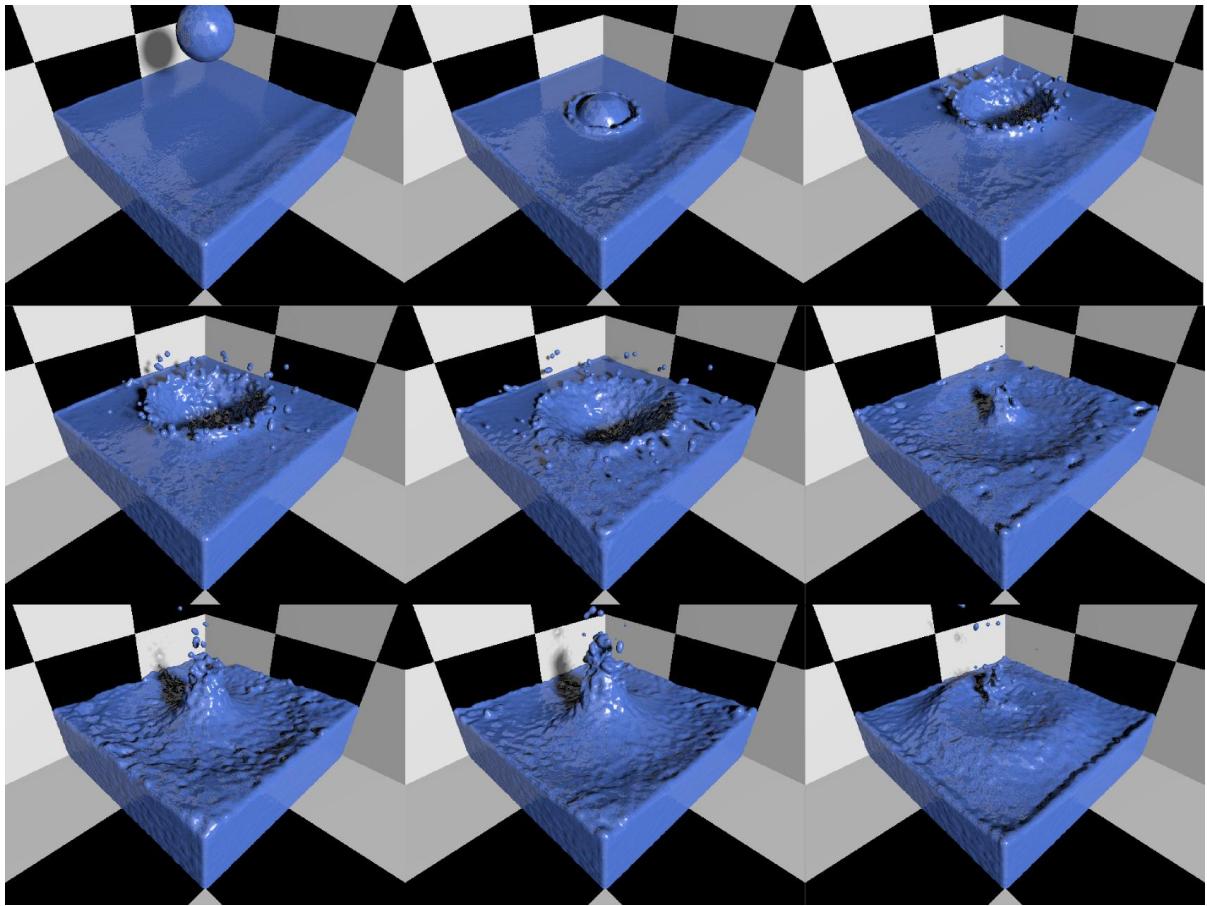


Figure 5.9: Time evolution of particle positions rendered in Povray showed at nine different time steps. The droplet falls into a basin of liquid and creates a characteristic impact crater.

It can be noticed that the new boundary formulation prevents boundary interactions successfully. The simulation does not include any surface tension effects which causes a slightly different behavior as observable in experiments (see Figure 5.10); nevertheless, the similarity between the photography and the upper right picture in Figure 5.9 is remarkable.

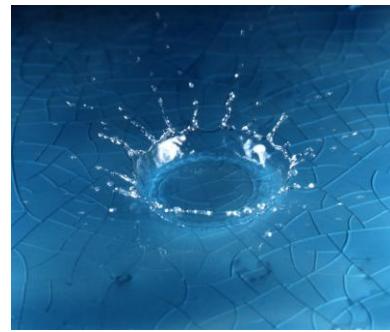


Figure 5.10: Photo of a “water crown” formed after the impact of a water droplet (source: DHD Multimedia Gallery).

An advantage of SPH over other methods is the intuitive handling of different fluids. These are simply discretized using SPH particles with different type-values. Figure 5.11 shows an example where a spherical drop of highly viscous liquid falls in a basin of lower viscous liquid. Both liquids drain eventually in a second chamber since the model includes a hole in the bottom that is opened at the onset of the simulation.

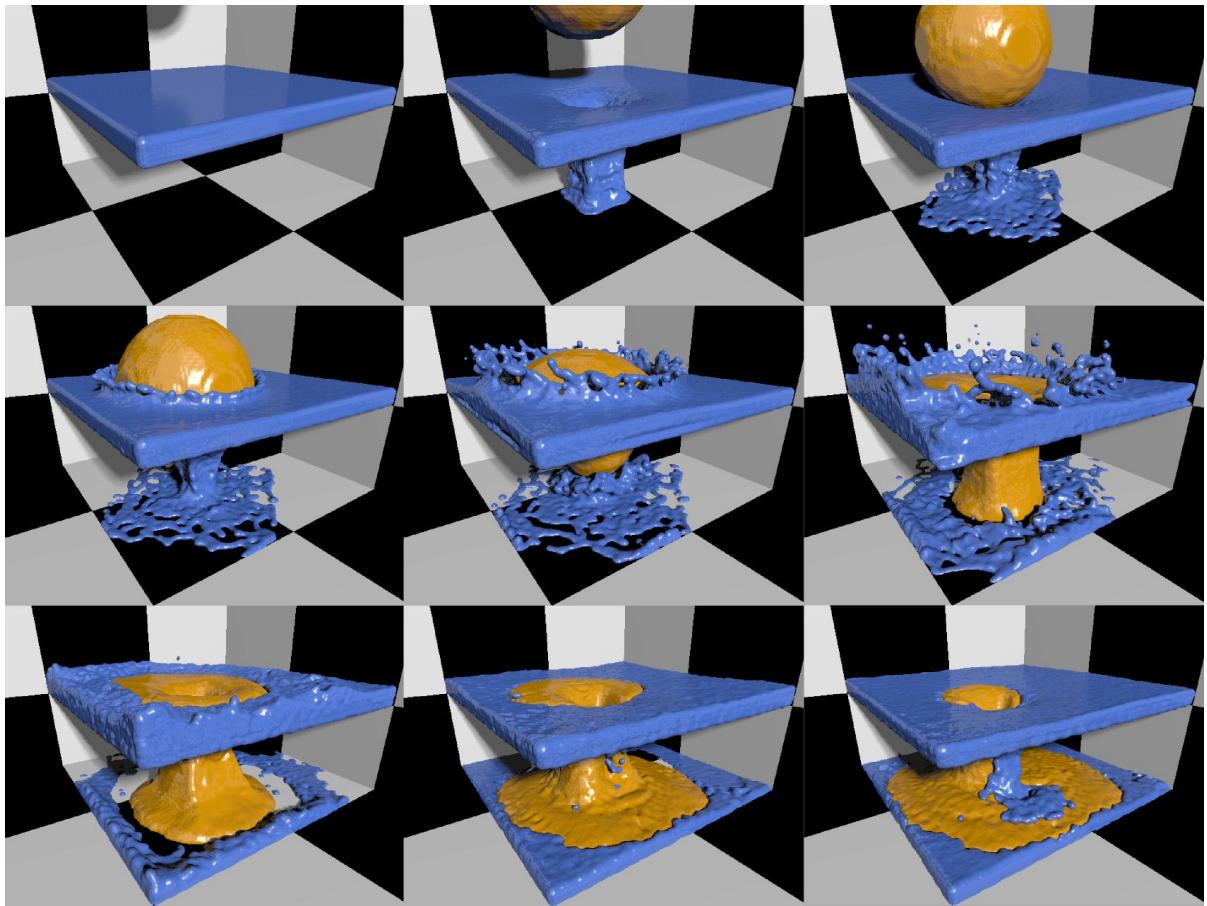


Figure 5.11: The interaction between fluids of different viscosities modeled using about 70,000 SPH particles.

The next simulation was intended to test the implementation of fluid structure interaction with rigid body dynamics. Using the approach outlined in section 5.2.2 a trough was modeled in Solid Works and subsequently transferred into a file containing particle positions and surface normal vectors. The file is loaded by the SPH simulation to define a body. For this test simulation the motion is limited to the rotation around one axis of the body. Figure 5.12 again shows a set of nine snapshots taken at different times.

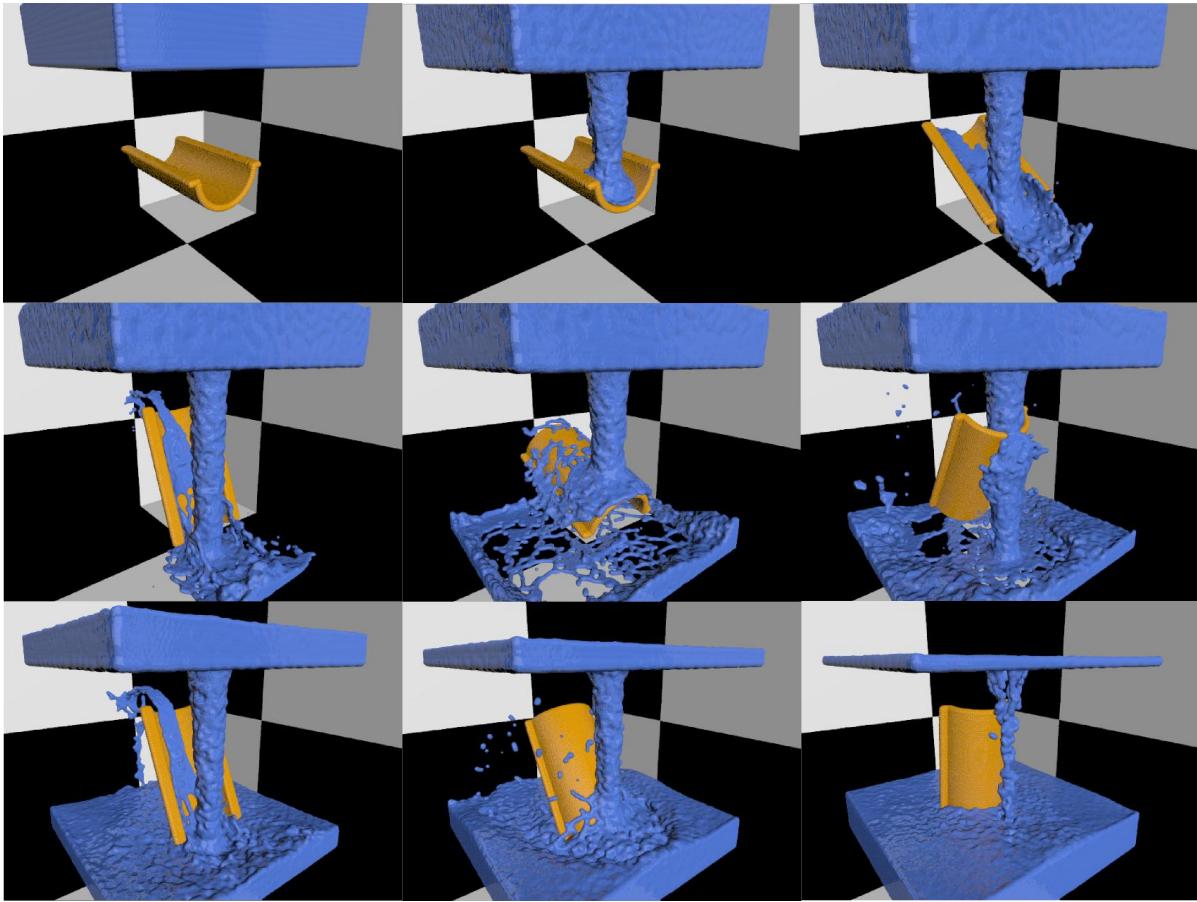


Figure 5.12: Fluid-structure interaction for the example of a trough that due to hydrodynamic forces starts to rotate around a pivot axis. The fluid is modeled with 250,000 SPH particles and the trough consists of 6760 boundary particles.

The simplified first principal boundary conditions used to calculate particle-wall as well as particle-structure interaction forces prevent boundary penetration successfully and lead to physically reasonable rigid body motion.

The hydrodynamic simulations above indicate that the implementation of the SPH algorithm works properly. They also show illustrate advantages of SPH. i.e. the capability to simulate different media in one framework and fluid-structure interaction can be included for both deforming and moving boundaries.

Chapter 6

Numerical Experiments

An extensive literature search led to the conclusion that to date SPH has only been used once in acoustic simulation to investigate the phenomenon of room reverberation [45].

The air in a room is discretized using a cuboid shaped lattice of SPH particles that is bounded by walls, modeled using repulsive forces according to [40]. A sound emitter, represented by a disc shaped boundary in the room is used to create sound waves that cause another disc shaped boundary, also located in the domain, to vibrate. The motion of this sound receiver is then analyzed by listening to a created output wav-file and also plotted in both time and frequency domain. It is reported that prior to the simulation the system has to reach an equilibrium state defined by a low level of kinetic energy. In the initial particle constellation the boundary forces acting on the fluid are not equal to forces due to pressure gradients. This is due to the chosen boundary formulation and results in high disturbances that need to be damped out in this first simulation procedure. Afterwards, wav-files are used to define the motion of the sound emitter that rudimentarily models a loudspeaker. A room of volume $8,000 \text{ m}^3$ is discretized using an unspecified number of SPH particles. Subsequently, different signals, specifically an impulse and different piano melodies, are used to define excitations. It is reported that the resulting audio file was perceived to sound more hollow and distant than the original sound excitation. Furthermore, a low pass filter effect with a break frequency of around $5,000 \text{ Hz}$ was noticeable in the frequency domain plot. It is also

reported that changed parameters lead to worse results and sometimes unstable results. Problems related to the boundary formulation can create strong disturbances that can corrupt results.

Unfortunately, no attempt is made to verify results with other simulation methods or analytical solutions. Besides, the effects of discretization or smoothing length on wave propagation are not analyzed in detail. In the following numerical experiments it will be shown that SPH is able to model sound propagation accurately (see section 6.1). The effect of important SPH parameters is analyzed using a set of one-dimensional simulations as shown in section 6.2. The applicability of different boundary formulations in acoustic SPH simulations will be addressed in section 6.3. The precision of the algorithm will be discussed in section 6.4 in relation to computational efficiency.

6.1 Two Dimensional Sound Propagation

This first numerical experiment aims to prove that SPH is capable of simulating sound wave propagation accurately without bulk flow effects. Therefore a simulation with 10,000 particles is set up into a square shaped domain. An initial velocity profile is imposed on the particles in order to create a pressure field whose time evolution can be observed. In order to create sound waves with very small pressure disturbances, the velocity of the particles is low in amplitude. The amplitude has to be small to ensure that the nonlinear dependence of the pressure on density in the equation of state Eq. (80) doesn't lead to differences compared to the linear wave equation. The pressure in the equation of state is nonlinearly dependent on the density because the internal energy is implicitly dependent on the density. The position dependent initial velocity function consists of two Gaussian distributions in x and y that are

multiplied with each other to create a symmetric pulse in the middle of the domain. The Cartesian components of the initial velocity excitation are given by the following equation,

$$v_x = v_y = 0.2 \cdot e^{-\left(\frac{|x|-0.09375}{0.013}\right)^2} \cdot e^{-\left(\frac{|y|-0.09375}{0.013}\right)^2}. \quad (114)$$

The ratio of specific heats is $\gamma = 1.4$ and initial values for density and specific internal energy are set to $\rho = 1 \text{ kg/m}^3$ and $u = 253.3 \text{ kJ/kg}$ respectively. According to Eq. (80) this leads to the standard atmospheric pressure of 101325 Pa . The theoretical wave speed in ideal gas at these conditions can be calculated with respect to the internal energy,

$$c = \sqrt{\gamma \cdot \frac{p}{\rho}} = \sqrt{\gamma(\gamma - 1)u} = 377 \text{ m/s}. \quad (115)$$

The artificial viscosity is set to zero to ensure that viscosity effects have no influence on the results of the SPH simulation, which are later compared to the solution of the linear wave equation. For simplicity, the particles are set up on a quadratic grid with equal particle spacing and no random component. This spares a highly damped preprocessing simulation where the particles can find their equilibrium positions [45]. Otherwise, the high noise level resulting from particle disorder makes acoustic simulations impossible. Boundaries are modeled using dynamic boundary particles as described in section 4.7.1. The positions of the particles in the three outer rows that frame the domain are therefore fixed and prevent the particles in the domain from dispersing. This is not exactly equivalent to a zero flux boundary condition because the fixed particles in the walls are compressible. Nonetheless, it is the easiest way of creating a boundary that prevents dispersing. The geometry and all simulation parameters are given in the following table.

Table 6.1: Parameters and geometry for the simulation of two-dimensional sound propagation.

Artificial viscosity parameters [29]:		
$\alpha = 0$	$\beta = 0$	$\gamma = 0.01$
equation of state (80): $p = (\gamma - 1)\rho u$		
$\gamma = 1.4$	$\rho_0 = 1 \text{ kg/m}^3$	$u_0 = 253.3 \text{ kJ/kg}$
smoothing length:		2.5 mm
smoothing kernel:		cubic spline
time step size:		$5.0e^{-7} \text{ s}$
number of particles:		10000
particle mass:		$3.52e^{-6} \text{ kg}$
particle spacing:		1.875 mm
width and depth of the gas square:		0.1875 m

The reference solution is obtained by a FDTD code that is also implemented in MATLAB. The MacCormack predictor-corrector method [56] is employed for time integration. Both simulations, the SPH and the FDTD, have the same spatial resolution and time step size. The same initial velocity excitation from Eq. (114) is used to compute initial flows between correspondent FD cells. Zero pressure Diriclet boundary conditions are employed for the two outer rows of FD cells and additional zero flow boundary conditions between the three outer rows of FD cells model rigid walls that bound the domain. Since the wave speed in the SPH simulation turned out to be slightly higher than the theoretical value of Eq. (115), the wave speed of the FDTD simulation is set to 388 m/s in order to create better comparable results. The deviation from the theoretical value of 377 m/s is 2.92% and a detailed analysis of this phenomenon follows in section 6.2.

Surface plots of the pressure field are created for each simulation method. The color corresponds to pressure (blue = low pressure; red = high pressure). The following surface plots show the pressure field at different time steps calculated with FDTD and SPH respectively. The left column shows the solutions obtained by FDTD and the right column

shows those obtained by SPH. The respective time is given at the top of each plot. The pressure axis, in z-direction, reaches from -15 Pa to 15 Pa in the FDTD plots and from 101310 Pa to 101340 Pa in the SPH plots. In contrast to SPH, where the absolute fluid pressure, p , is used, results of FDTD simulations are typically given in the acoustic pressure, $\tilde{p} = p - p_{mean}$.

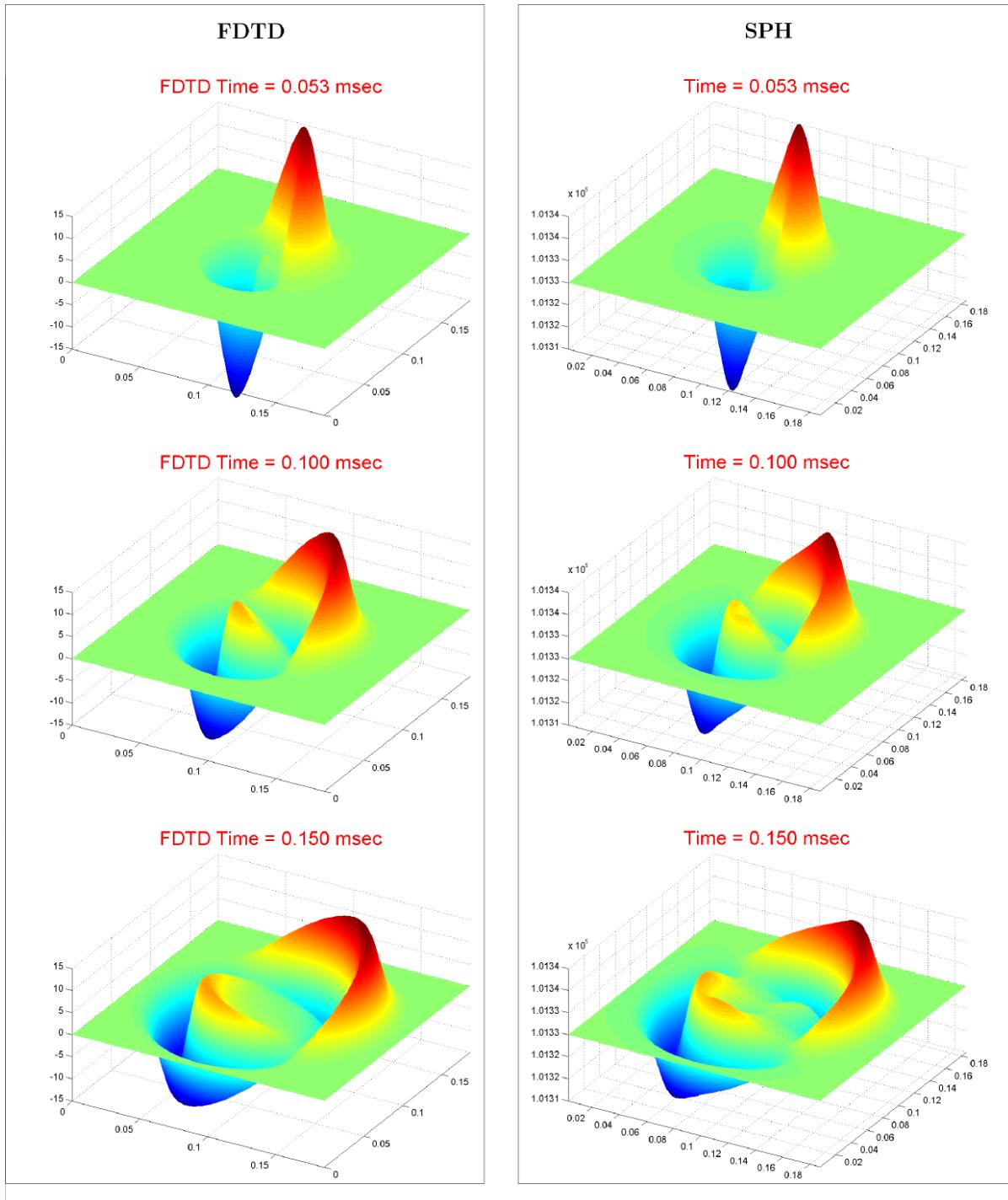


Figure 6.1: Surface plots of the pressure field obtained by FDTD (left column) and SPH (right column) for time steps 0.053 ms, 0.100 ms and 0.150 ms.

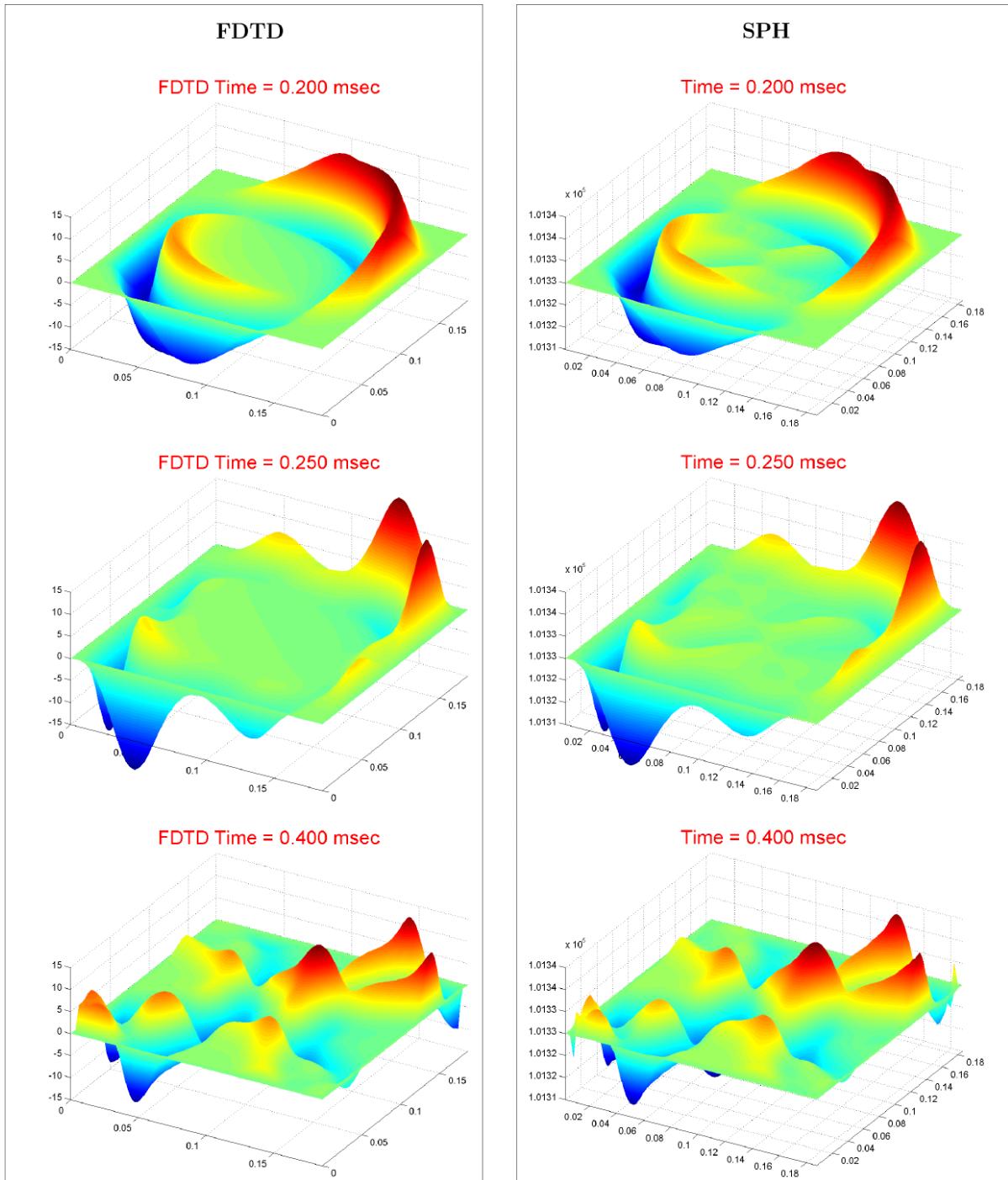


Figure 6.2: Surface plots of the pressure field obtained by FDTD (left column) and SPH (right column) for time steps 0.200 ms, 0.250 ms and 0.400 ms.

The slightly adapted wave speed for the FDTD simulation leads to a good agreement of the plots for all time steps. Therefore it can be assumed that the wave speed in this SPH

simulation is equal to 388 m/s which is only slightly different from the theoretical value.

It can be seen from the figures above that the shape of the pressure surface plots is also in good agreement with the FDTD solution. Only small shape and amplitude differences are noticeable. The shape of the main disturbances in the SPH simulation shows additional curvature of the edges which can best be seen at a time of 0.100 ms in Figure 6.1. The second obvious difference consists of artifacts of the velocity excitation in the middle of the domain that lead to additional disturbances. These can best be seen at time steps 0.200 m/s and 0.250 ms in Figure 6.2.

One source likely causing these shape differences is the grid-like structure of the particle positioning leading to anisotropic behavior. Furthermore amplitude differences can be caused by the bulk flow due to the excitation (taken into account in the SPH but not in the FDTD simulation). However, this effect is small for the very short simulation time involved. Consequently, no considerable amount of bulk flow can build up.

Another result of the simulation is that the boundary formulation with fixed boundary particles in the SPH simulation is very close to a rigid wall like formulated in the FDTD simulation. This conclusion can be drawn because the time evolution of the pressure field stays similar after the waves hit the walls and get reflected. This can be seen especially well in the plots for 0.250 ms and 0.400 ms (see Figure 6.2).

6.2 The Smoothing Length and its Impact on Wave Speed

It is known that the shape of the smoothing kernel and the smoothing length crucially influence the accuracy of SPH simulation results [57]. A parametric study with 1-D

simulations is set up to investigate the effect of different smoothing lengths on the wave speed in acoustic SPH simulations.

A number of 100 SPH particles are equally spaced along a line. At the beginning of the simulation all particles are at rest and under standard pressure. Additional boundary particles are defined at both the ends that prevent the gas particles from dispersing. Mirroring particles according to section 4.7.3 are chosen since this is straightforward to implement for one-dimensional problems and it models rigid wall with a higher quality than dynamic boundary particles.

A simple boundary and initial value problem, Eq.(116), is chosen so that an analytical solution to the wave equation can be determined.

$$u_{tt} = c^2 v_{xx} \quad | x \geq 0; t \geq 0 \quad (116)$$

$$\begin{aligned} ICs : v(x, 0) &= 0; \quad v_t(x, 0) = 0 \\ BC : v(0, t) &= f(t) = v_0 \sigma(t) \end{aligned}$$

Subscripts indicate derivatives with respect to time, t , or space, x , and σ is a step function at the origin. The particles are excited from the left side through a constant velocity excitation, v_0 , that starts at time $t = 0$. An analytical solution for the particle velocity can be derived using the Laplace transform [58] leading to the following result,

$$v(x, t) = f(t - x/c) = v_0 \sigma(t - x/c). \quad (117)$$

This means that a step function with velocity amplitude v_0 runs into the domain starting at the left boundary. According to the adiabatic process equation, the pressure, p , is related to the density, ρ , as,

$$\frac{p}{p_0} = \left(\frac{V_0}{V}\right)^\gamma = \left(\frac{\rho}{\rho_0}\right)^\gamma. \quad (118)$$

This equation can be linearized since the density variations are very small. An expression for the density is then given as,

$$\rho = \frac{\rho_0 p}{\gamma p_0} + \rho_0 . \quad (119)$$

The relation between the velocity, v , and the pressure, p , can be obtained combining the continuity equation,

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0 , \quad (120)$$

and Eq.(119) as follows,

$$\frac{\rho_0}{\gamma p_0} \frac{\partial p}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0 . \quad (121)$$

The assumption of small density variations simplifies this equation to,

$$\frac{1}{\gamma p_0} \frac{\partial p}{\partial t} + \frac{\partial v}{\partial x} = 0 . \quad (122)$$

The partial derivative of the velocity, v , with respect to space, x , can be calculated according to Eq.(117),

$$\frac{\partial v}{\partial x} = \frac{v_0}{c} \delta(t - x/c) , \quad (123)$$

where δ is the Dirac delta function. Time integration of Eq.(122) leads to the following expression for the pressure with respect to position and time,

$$p = p_0 + \frac{\gamma p_0 v_0}{c} \sigma(t - x/c) . \quad (124)$$

Like in Eq.(117), this is a step function (see Figure 12), different only in amplitude. The chosen experiment has the advantage that the wave speed can be calculated using the simulation results for the pressure.

$$c = \frac{\gamma p_0 v_0}{p_{high} - p_0}, \quad (125)$$

Where p_{high} is defined as the pressure on top of the step function. Its value is approximated by averaging the pressures for the first 20 SPH particles at the left end of the domain. Simulation results at $t = 0.2 \text{ ms}$ are reported, which ensures that the step function has enough time to travel into the domain. A set of 41 SPH simulations with different smoothing lengths is carried out. The simulation parameters are given in the following table:

Table 6.2: Parameters and geometry for the one-dimensional simulation.

Artificial viscosity parameters [29]:		
$\alpha = 0$	$\beta = 0$	$\gamma = 0.01$
equation of state (80): $p = (\gamma - 1)\rho u$		
$\gamma = 1.4$	$\rho_0 = 1 \text{ kg/m}^3$	$u_0 = 253.3 \text{ kJ/kg}$
smoothing lengths:		(1 : 0.1 : 5) mm
smoothing kernel:		cubic spline
time step size:		$5.0e^{-7} \text{ s}$
number of particles:		100
particle mass:		$1e^{-3} \text{ kg}$
particle spacing:		1 mm
length of the “gas pipe”:		0.1 m
velocity excitation v_0		$2e^{-3} \text{ m/s}$

The following figures show the pressure profiles after 0.2 ms for a collection of simulations corresponding to smoothing lengths of 1 mm, 1.5 mm, 1.7 mm, 2 mm, 3 mm and 5 mm respectively.

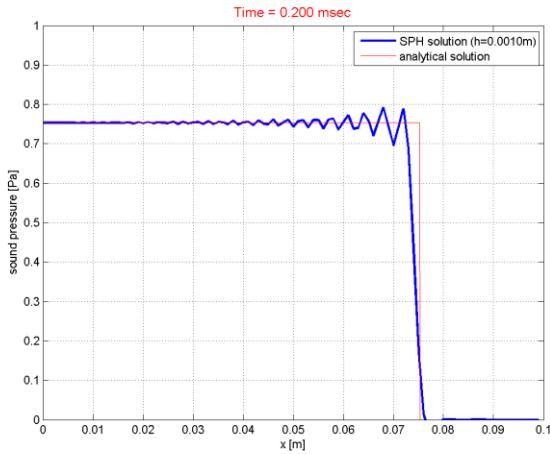


Figure 6.3: Sound pressure plot for smoothing length equal 1 mm. Very good agreement with the analytical solution.

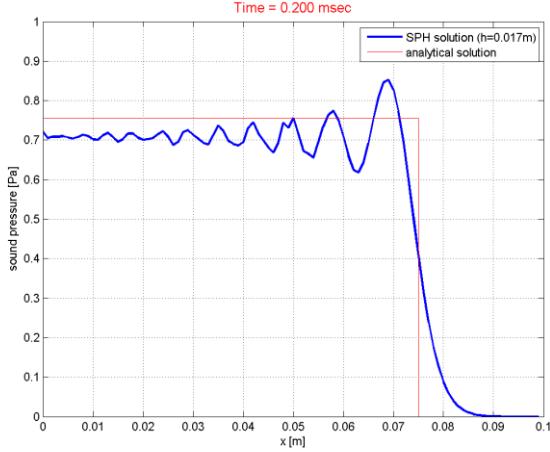


Figure 6.4: Sound pressure plot for smoothing length equal 1.7 mm. Wave speed is slightly higher compared to the analytical solution.

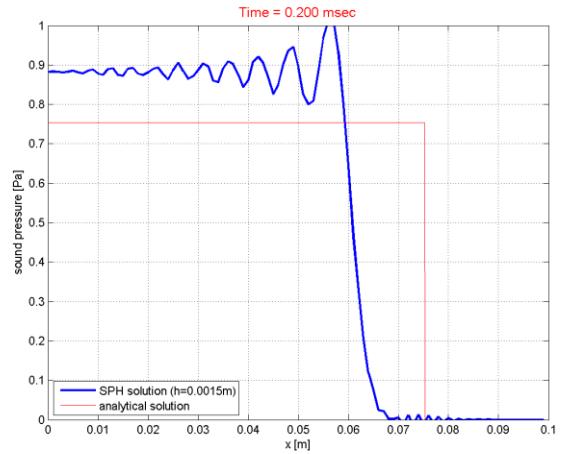


Figure 6.5: Sound pressure plot for smoothing length equal 1.5 mm. The wave speed is lower than the theoretical value.

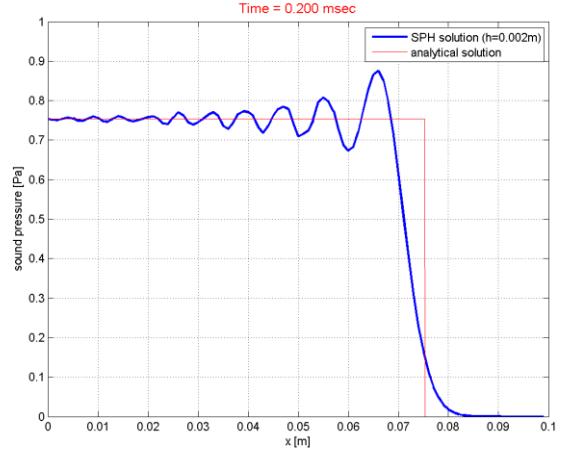


Figure 6.6: Sound pressure plot for smoothing length equal 2 mm. The wave speed agrees with the analytical solution.

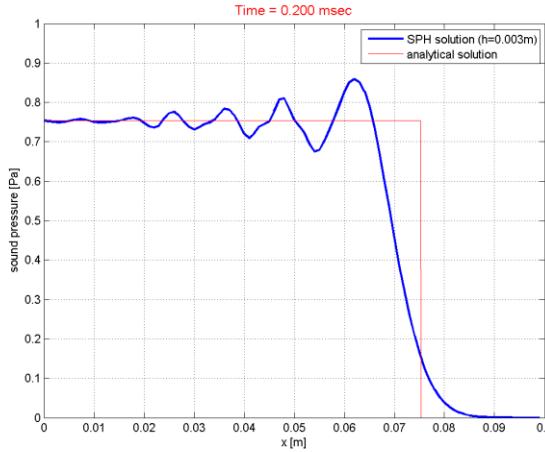


Figure 6.7: Sound pressure plot for smoothing length equal 3 mm. The sound speed agrees with the analytical solution but the quality of the numerical solution begins to suffer.

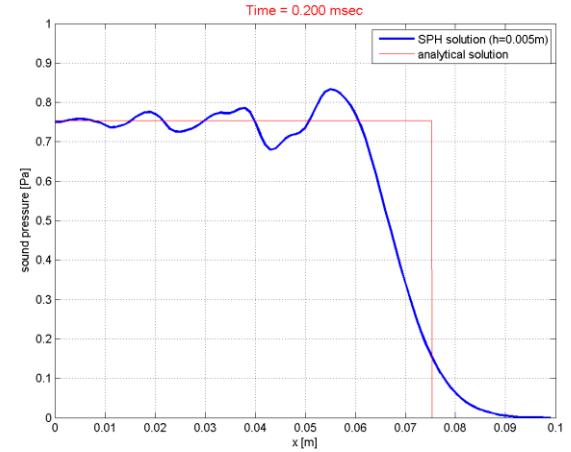


Figure 6.8: Sound pressure plot for smoothing length equal 5 mm. The sound speed agrees with the analytical solution but the quality of the numerical solution is poor.

The wave velocity in each simulation is calculated according to Eq. (125) and plotted over the smoothing length as shown in Figure 6.9.

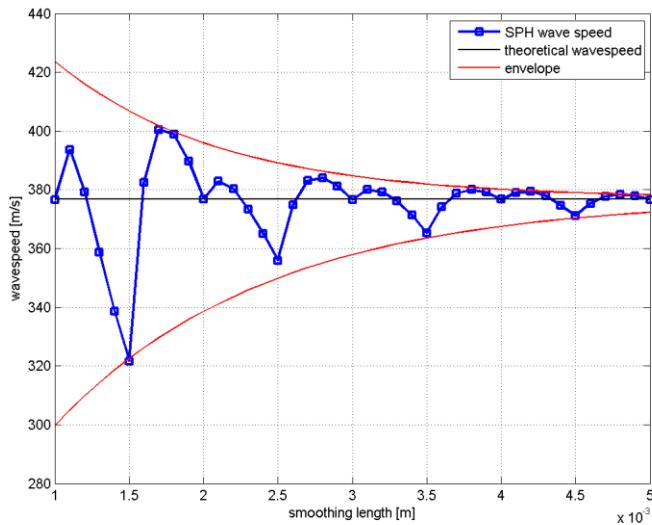


Figure 6.9: Wave speeds of all 41 SPH simulations plotted over the smoothing length. Convergence of the SPH wave speed towards the theoretical value can be observed with growing smoothing length.

The relative wave speed error can be plotted over the smoothing length, adimensionalized using the particle spacing as shown in Figure 6.10.

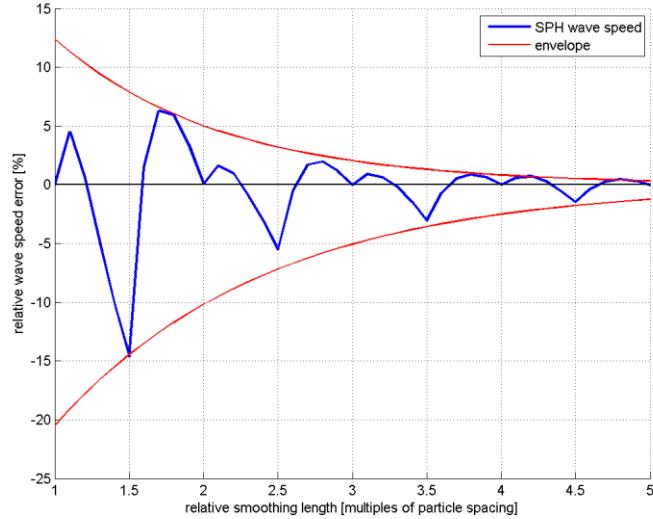


Figure 6.10: Adimensionalized version of Figure 6.9. The smoothing length divided by the particle spacing is plotted on the x-axis and the relative error in the wave speed is plotted on the y-axis.

The correlation shown in Figure 6.10 applies to all one-dimensional SPH simulations with equal particle spacing and cubic spline kernel. It confirms that the results are significantly more accurate (in particular better propagation speeds) if the smoothing length, h , is equal to integer multiples of s and least accurate with $h = 1.5s$, where s is the particle spacing [57]. Therefore, unstructured particle positioning always results in locally varying wave speeds. In two-dimensional simulations with grid-like particle positioning the wave speed varies with the orientation with respect to the grid. This explains the shape differences in the two-dimensional experiment above.

Furthermore Figure 6.10 shows that the propagation speed becomes more robust for higher smoothing lengths. For one-dimensional simulations with unstructured particles and also for two-dimensional simulations the curves do not look identical but will nevertheless

match the trend. The difference is due to the particle approximation which differs between the one-dimensional case, where particles are placed on a straight line, and the two-dimensional case, where particles are arranged in a plane. Even though the wave speed will show significant changes locally, for simulations with disordered particles it can be assumed that these local changes don't affect the global wave speed. Lower and higher local wave speeds average out provided the discretization is sufficiently fine.

Looking at Figures 6.3 to 6.8 another trend is noticeable. The SPH approximation quality decreases with higher smoothing lengths. Apparently, high smoothing lengths tend to filter out high frequency components of the signal. In this experiment the filtering effect causes a smoothed out edge and lower frequency deviations from the ideal step function. Nonetheless a comparison to a one-dimensional FDTD simulation of the same experiment and with the same discretization length shows that SPH results calculated with low smoothing length can lead to competitive solutions (compare Figures 6.3 and Figure 6.11).

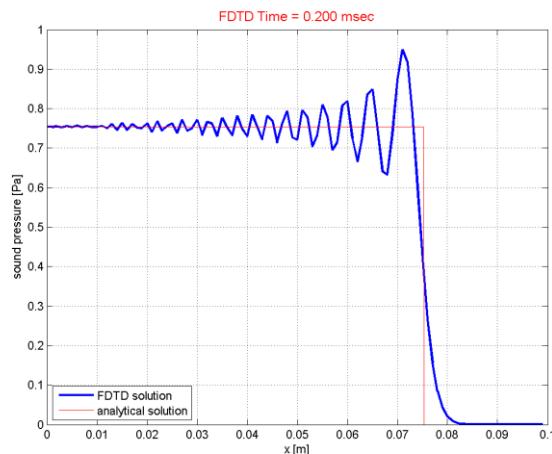


Figure 6.11: FDTD solution for the same experiment. The wave speed agrees with the analytical solution.

Other SPH simulations indicate that waves with small wave lengths tend to get “smoothed out”, meaning that the amplitude decreases while the wavelength increases. It is reasonable to define a lower limit for the wave lengths in order to prevent a considerable amount of “smoothing”. A series of tests, [57], has shown good agreement with theory when the wavelength, λ , satisfies the condition $\lambda \geq 2\pi h$, where h is the smoothing length.

6.3 Sound excitation and propagation in a Tube

It has been shown that SPH is capable of modeling sound propagation accurately and a major influence factor on wave speed has been identified. To investigate both sound propagation in three dimensions and sound excitation through moving boundaries a simple tube experiment similar to that of section 6.2 was chosen. Figure 6.12 demonstrates the geometry of this three-dimensional experiment and the chosen coordinate system.

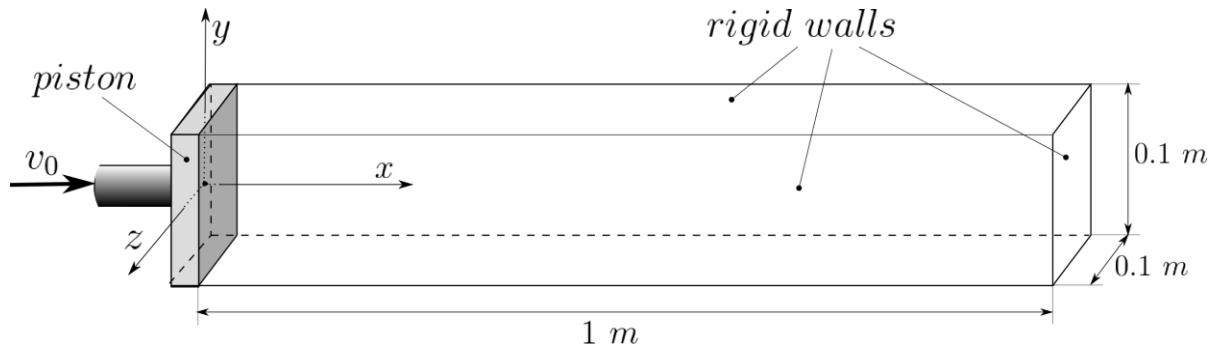


Figure 6.12: Geometry and coordinate axis of the tube. It has a quadratic cross section and is filled with ideal gas.

The piston at the left end of the tube moves with a constant velocity leading to the problem formulation given in Eq. (116). The only difference to the one-dimensional experiment of section 6.2 is that the full three-dimensional complexity has to be resolved, meaning that no assumptions regarding symmetry are made and all the walls have to be

modeled as such. The fact that the problem still is symmetric in y and z direction is useful for the analysis of simulation results. It means that field variables are expected to be constant over the cross section of the tube. However, this is only true if the boundary formulations model the side walls of the tube accurately.

In contrast to the Matlab program, the three-dimensional implementation on the GPU uses only single precision for all calculations. This is due to current hardware limitations and is the source of accuracy issues. Specifically, in the case of the SPH implementation, it causes simulations with gas particles to become unstable. Other GPUs can perform double precision arithmetic, however this slows down the computation by about a factor of four. For this reason another solution to this problem was investigated. The most likely source of accuracy problems can be found in the summations of the SPH formulation (see Eqs. (64), (69) and (77)). Given that no stability problems were observed for liquid simulations the equation of state for ideal gas has been modified slightly which improved the stability drastically (see Eq.).

$$p = (\gamma - 1)(\rho - \rho_{mean})(u - u_{mean}) \quad (126)$$

Similar to the equation of state for liquids (see Eq.(81)) this formulation returns zero pressure for initial conditions, $\rho = \rho_{mean}$ and $u = u_{mean}$. On the other hand it does not change the theoretical result of the SPH summations, Eqns. (64), (69) and (77).

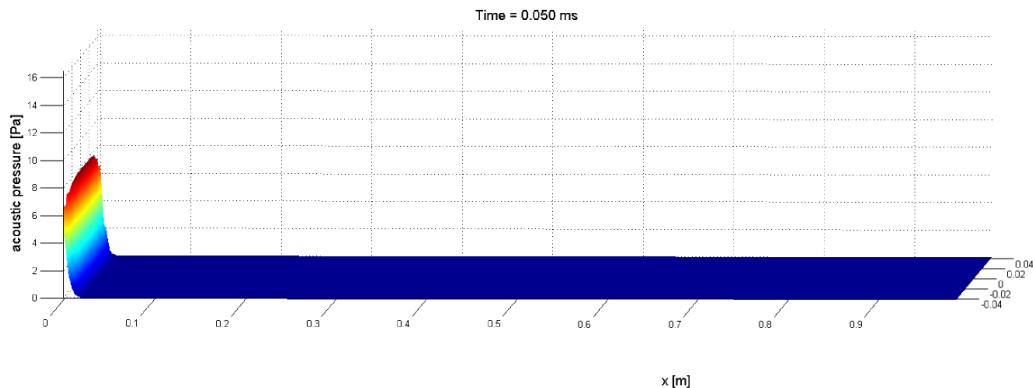
In order to assess and compare different boundary formulations, the tube, shown in Figure 6.12, is modeled using the three-dimensional GPU implementation of the SPH algorithm and boundary formulation described in section 4.7.

In the first simulation the boundaries are modeled using dynamic boundary particles. These are arranged around the SPH gas particles which form the problem domain. SPH particles are placed in a cubic lattice in order to obtain an initial constellation that is in equilibrium. Boundary particles of the left wall at $x = 0$ move with a constant velocity of $v_0 = 0.02 \text{ m/s}$ in order to model the piston. The parameters employed in the following simulations are given in Table 6.3.

Table 6.3: Parameters and geometry for the three-dimensional tube simulations.

Artificial viscosity parameters [29]:		
$\alpha = 0$	$\beta = 0$	$\gamma = 0.01$
equation of state: $p = (\gamma - 1)(\rho - \rho_{mean})(u - u_{mean})$		
$\gamma = 1.4$	$\rho_0 = 1 \text{ kg/m}^3$	$u_0 = 253.3 \text{ kJ/kg}$
smoothing length:		4.0 mm
smoothing kernel:		cubic spline
time step size:		$1.0e^{-6} \text{ s}$
number of particles:		270,000
particle mass:		$3.70e^{-8} \text{ kg}$
particle spacing:		3.33 mm
geometry of the tube:		$1 \text{ m} \times 0.1 \text{ m} \times 0.1 \text{ m}$
velocity excitation v_0		$2e^{-2} \text{ m/s}$

Figure 6.13 shows the simulation results for the pressure in the tube on a horizontal level at $y = 0 \text{ m}$.



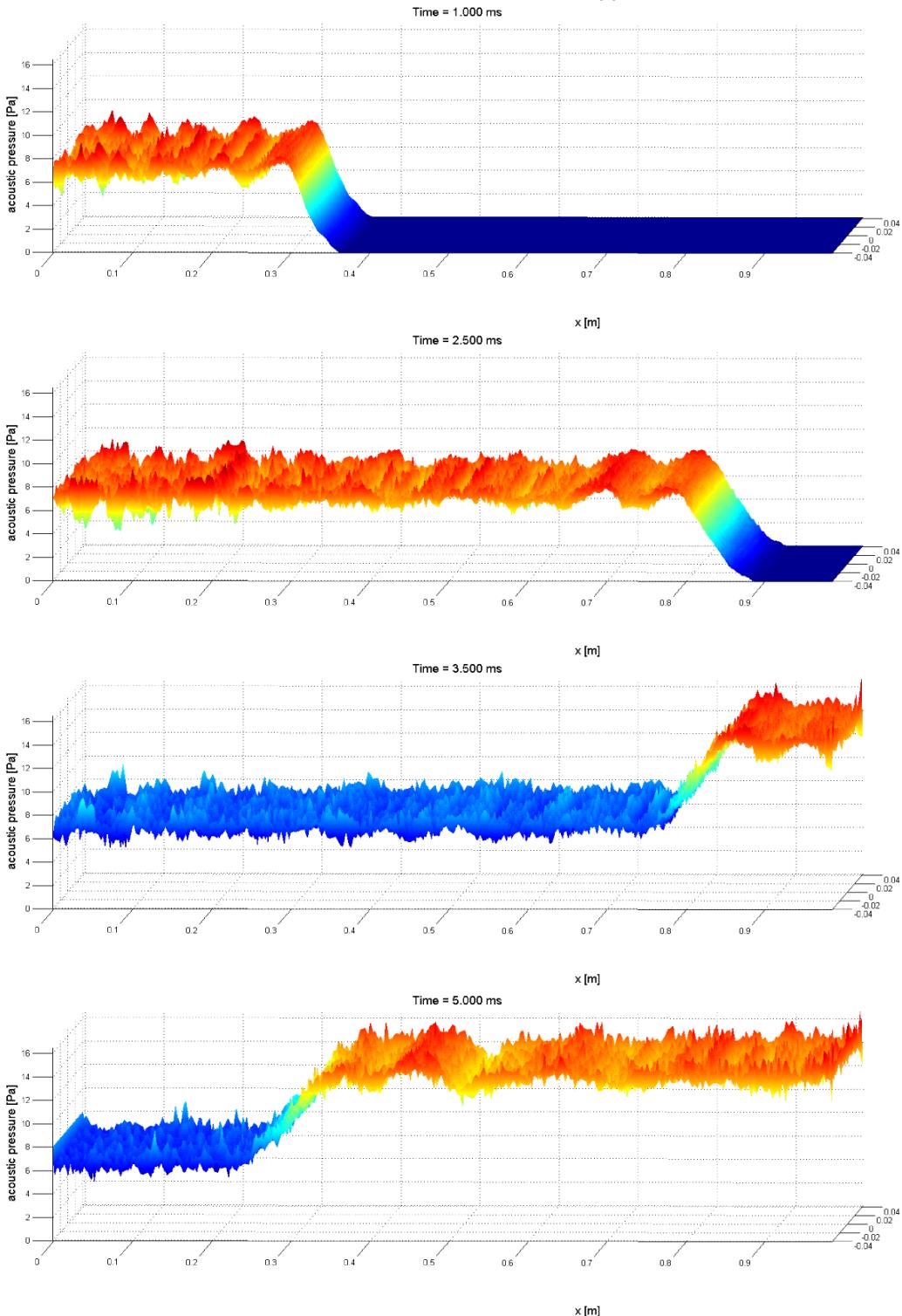


Figure 6.13: Pressure plotted for five subsequent times (given at the top of each plot) over the position on a horizontal level at $y = 0$ m. The wave, excited by piston motion, travels form the left end of the tube to the right and gets reflected. Dynamic boundary particles are used.

Similar to the experiment demonstrated in section 6.2, a discontinuity with a constant level is expected to propagate through the tube doubling its height when reflected at the right end of the tube. It can be shown that the pressure level is reproduced with good accuracy. However, it can also be seen that, along with the low frequency fluctuations known from section 6.2, a high level of noise is introduced by the chosen boundary formulation.

The use of mirror particles has the potential to improve this problem because the zero-flux boundary conditions, given in this simulation, can be represented exactly. Initial simulations revealed a problem related to particle deficiency as explained in section 4.7.3. Different than above, in the simulation with dynamic boundary particles, no particles get created on sharp edges as illustrated in Figure 6.14.

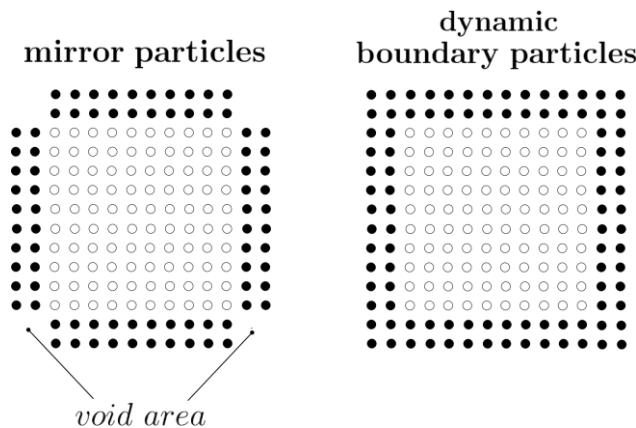


Figure 6.14: This graphic shows a representation of the tube cross section. Left: Particle deficiency at sharp corners when using the mirror particle approach; Right: No particle deficiency due to the use of dynamic boundary particles.

This results in opposing forces created through mirror particles which are insufficient for particles located near a sharp edge. This leads to pressure loss since these SPH particles start to move towards the edges. This effect is illustrated by a simple experiment in which the tube

is initially under constant pressure. Figure 6.15 shows the pressure distribution in the cross section of the tube for three subsequent time steps.

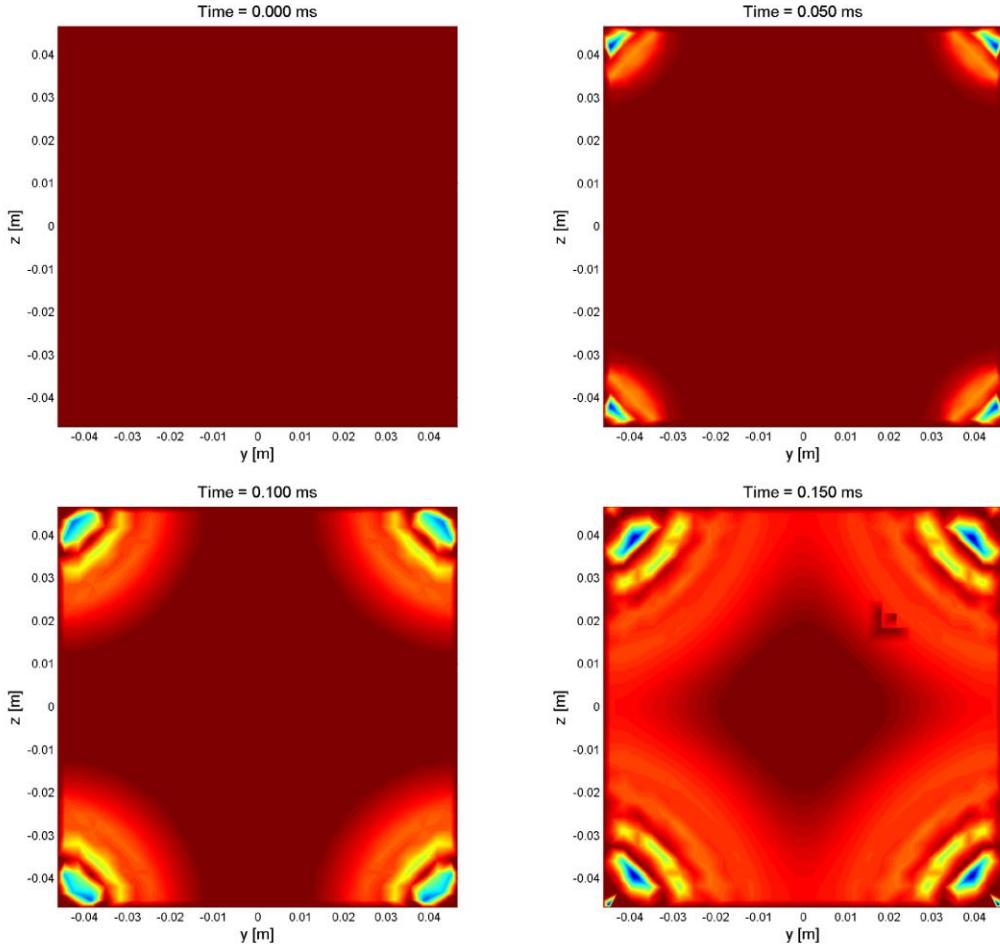


Figure 6.15: Pressure loss at sharp corners when using mirror particles. The plots show the pressure (represented as color) distribution over the cross section of the tube for four subsequent time steps.

Due to particle motion towards sharp edges, the pressure decreases at the corners. This problem can be solved using patches of dynamic boundary particles that fill the voids area of mirror particles as shown in Figure 6.16.

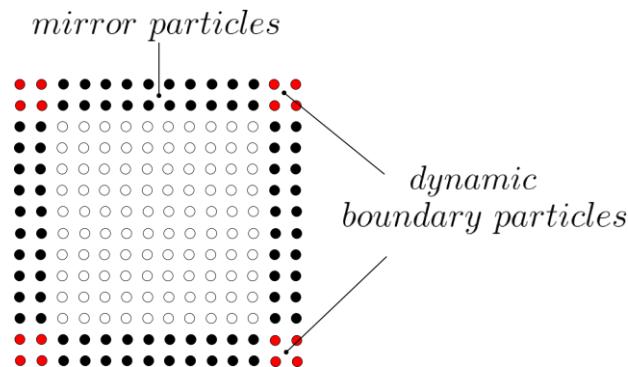
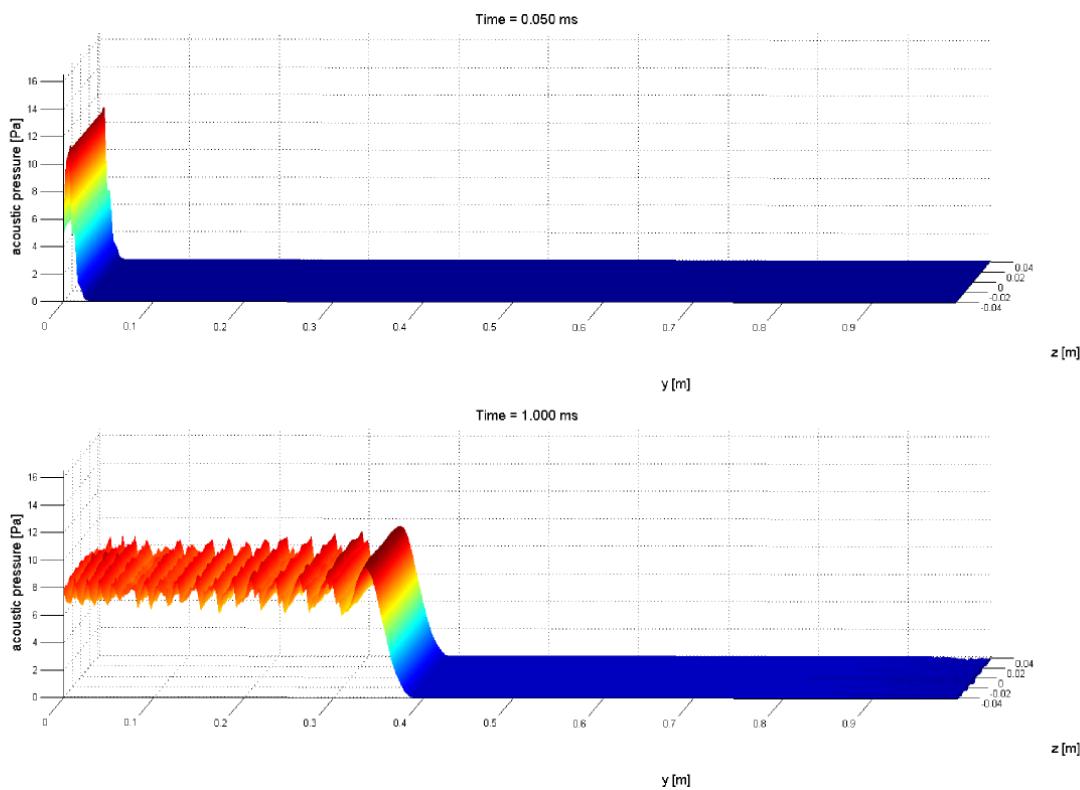


Figure 6.16: Patches of dynamic boundary particles are placed in the corners to prevent particle deficiency.

The results of the simulation which uses both mirror particles and dynamic boundary particles are demonstrated in Figure.



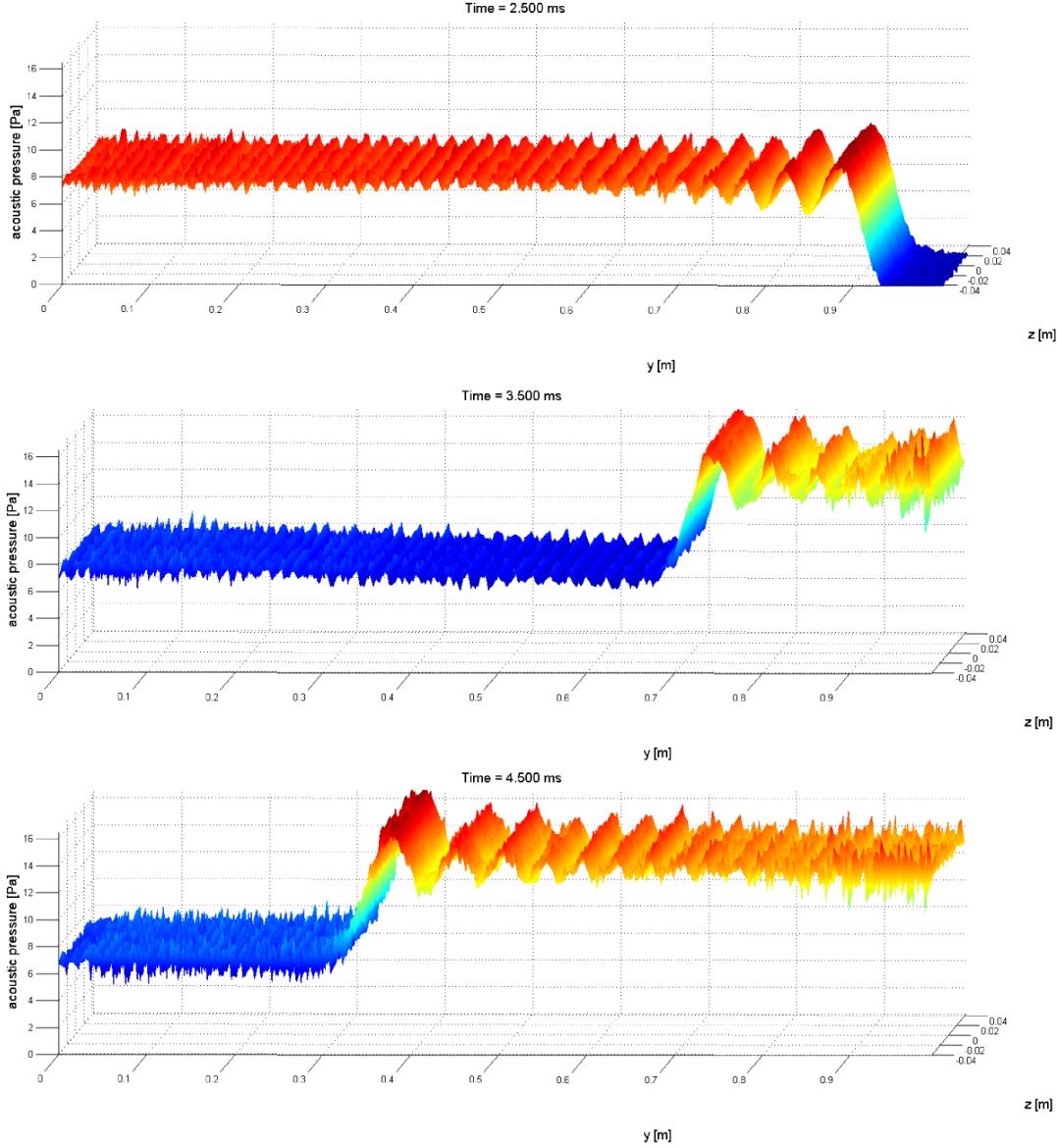


Figure 6.17: Pressure plotted for five subsequent times (given at the top of each plot) over the position on a horizontal level at $y = 0$ m. The wave, excited by piston motion, travels form the left end of the tube to the right and gets reflected. A combination of mirror and dynamic boundary particles is used to describe the rigid walls.

While the pressure level still shows good agreement with the expected solution, it can be noticed that the amount of high frequency noise is reduced (see). However, in the last two plots ($time = 3.5\text{ ms}$ and $time = 4.5\text{ ms}$) the noise level is again very high. It can also be observed that the wave has a different shape as in Figure 6.13. the deviations from the

analytical solution are higher in both amplitude and frequency. Furthermore, the edge of the incontinuity is steeper and does not get smoothed out as much as in Figure 6.13 even though all parameters are identical for both simulations.

Tests with repulsive forces as a boundary formulation indicate that they are not well suited for acoustic simulations. High disturbances, due to unbalanced fluid and boundary forces at the initial arrangement, need to be damped out before a reasonable simulation can start. However, it turned out that the pressure distribution after the first step is very inhomogeneous. High pressure peaks remain after the kinetic energy has been dissipated from the system. Attempts to obtain a homogeneous initial pressure distribution did not lead to promising results. Simplified first principle boundary conditions, which were tested as well, also suffer from the problem. The slightly increased boundary pressure, necessary to prevent boundary penetration, has the same effect as repulsive forces. The advantages of these very flexible and relatively straight forward boundary formulations can only be exploited if a way can be found to obtain a constant initial pressure distribution.

In conclusion, the mirror particle approach lead to the best results even though there are problems with sharp edges. These can, in the specific case of the rectangular tube, be solved by using patches of dynamic boundary particles. On the other hand this procedure is tedious and cannot be applied in every case. A solution to this problem could be to smooth out edges. If the use of mirror particles leads to good results with relatively flat surfaces it can be assumed that the error induced by the boundary formulation reduces with increased discretization. For this reason a simple experiment with a tube of circular cross section is

conducted in which the tube is initially under a constant pressure. Figure 6.18 shows the time evolution of the pressure profile in the cross section.

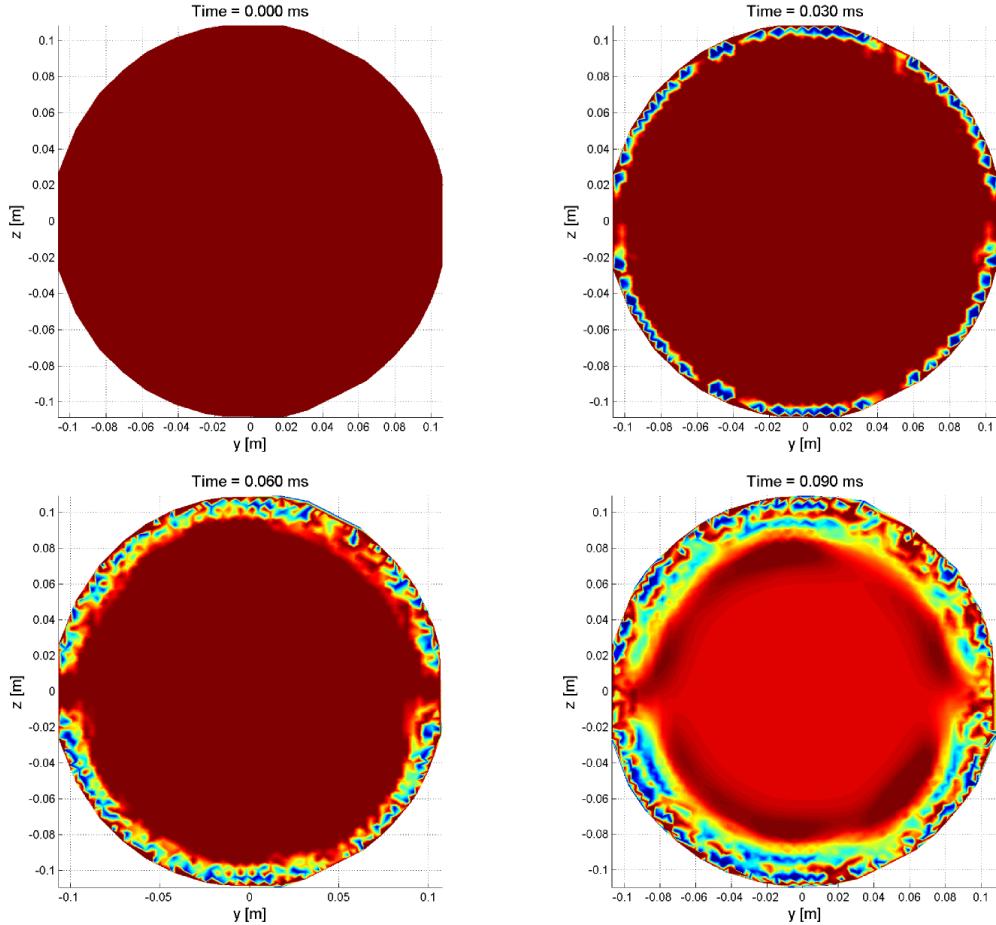


Figure 6.18: Pressure loss at the surface of a tube with circular cross section. The mirror particles produce disturbances and do not prevent boundary penetration.

Experiments with differently sized tubes led to the conclusion that significant disturbances are created even for large radii.

6.4 Work Precision Diagram

Even though the computational efficiency has been dramatically increased, leveraging the computing capabilities of GPUs, it is still important know which resolution is required in each specific simulation. Especially when dealing with in three-dimensional problems, too fine resolution leads to a very high number of SPH particles and thus to long simulation time. This is due to the fact that the total number of particles increases cubically with resolution. Conversely, too coarse discretization is typically results in poor accuracy. For this reason it will be discussed how the three-dimensional SPH implementation, explained in section 5.2, scales with the number of particles and which level of accuracy can be expected for a given discretization quality.

A simple experiment is set up, in which the sound propagation is simulated using SPH. The results are then compared to a reference solution, calculated using FDTD. A cubical lettuce of SPH particles is used to discretize the gas in a cube with one meter side length. An initial velocity excitation, similar to that in section 6.1, creates a pressure wave in the center of the cube. The concentric sound wave propagation is simulated for $8e^{-4}$ seconds, which ensures that the wave front does not reach the boundary. Therefore, if using the equation of state in form of Eq. (126), no explicit boundary formulation needs to be employed which automatically ensures that boundary formulations cannot corrupt simulation results.

Simulation results after $8e^{-4}$ seconds are given in Figure 6.19 for increasing resolutions, starting from 26, in steps of 25, up to 151 particles per meter. The plot shows the pressure distribution in a horizontal slice at $z = 0 \text{ m}$.

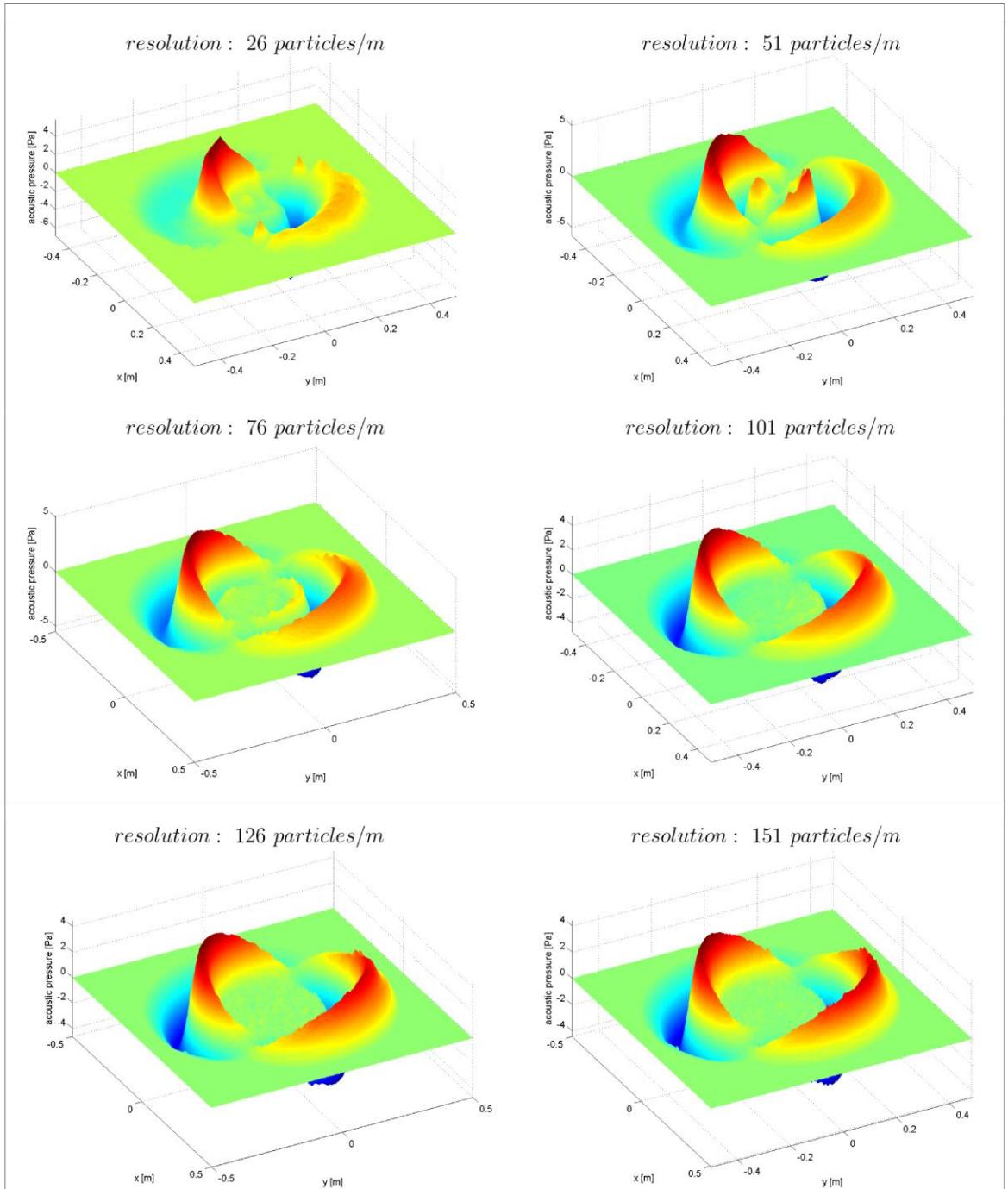


Figure 6.19: Pressure over position after $8e^{-4}$ seconds in a horizontal slice at level $z = 0$ m. The resolution is given in particles per meter on top of each plot.

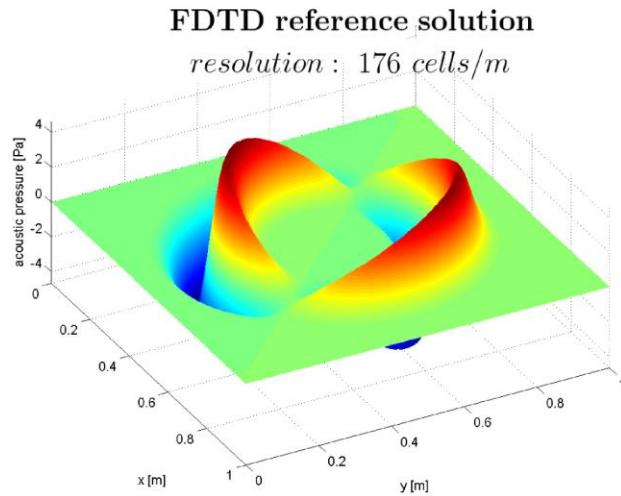
Table 6.4: Parameters and geometry for the simulations.

Artificial viscosity parameters [29]:		
$\alpha = 0$	$\beta = 0$	$\gamma = 0.01$
equation of state: $p = (\gamma - 1)(\rho - \rho_{mean})(u - u_{mean})$		
$\gamma = 1.4$	$\rho_0 = 1 \text{ kg/m}^3$	$u_0 = 253.3 \text{ kJ/kg}$
smoothing length, h :		$1.2\Delta s$ (Δs =particle spacing)
smoothing kernel:		cubic spline
geometry of the cube:		$1 \text{ m} \times 1 \text{ m} \times 1 \text{ m}$

The following three-dimensional initial velocity excitation was used:

$$v_x = v_y = v_z = 0.2 \cdot e^{-(|x|/0.06)^2} \cdot e^{-(|y|/0.06)^2} e^{-(|z|/0.06)^2}. \quad (127)$$

It can be noticed that, dependent on the discretization quality, peaks and especially the center areas show significant shape and amplitude variations. As already mentioned, a reference solution is calculated using the same FDTD implementation, described in section 6.1. The result of this reference simulation is illustrated in Figure 6.20, where the pressure at level $z = 0 \text{ m}$ is plotted over position.

**Figure 6.20: Reference solution obtained using FDTD. The resolution is here 176 cells per meter.**

A set of 1,000 positions, distributed on a cubical lattice over the whole three-dimensional problem domain, is chosen for which the relative error to the FDTD solution is calculated. The arithmetic mean value of these error terms, plotted over the resolution can be seen in Figure 6.21.

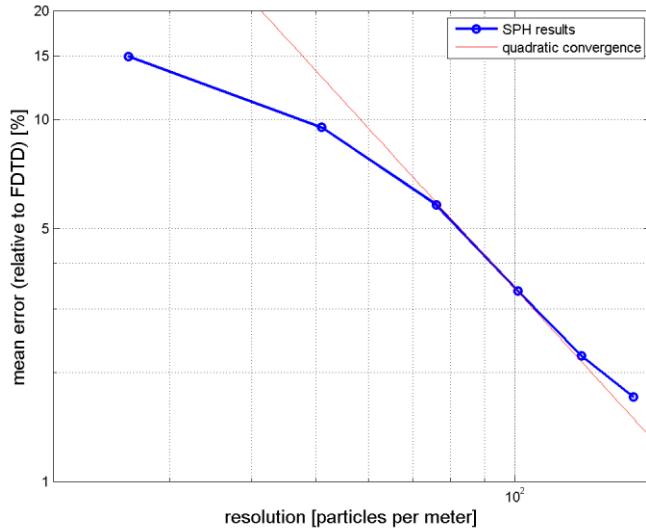


Figure 6.21: Arithmetic mean of the relative error, measured at 1,000 different positions in the domain.

It can be noticed that the SPH results converge with increasing resolution towards the FDTD solution. For increasing resolution, the SPH results show quadratic convergence. The two simulations with the lowest discretization show less error than theoretically predicted but it can clearly be seen that the results have significant deficits (see Figure 6.19). The increased error for the simulation with the highest discretization can be traced back to the fact that the wave speed differs slightly between the SPH and the FDTD simulation.

The time required to execute one simulation step is reported in Figure 6.22 for a NVIDIA GeForce 8800 GTX graphic card.

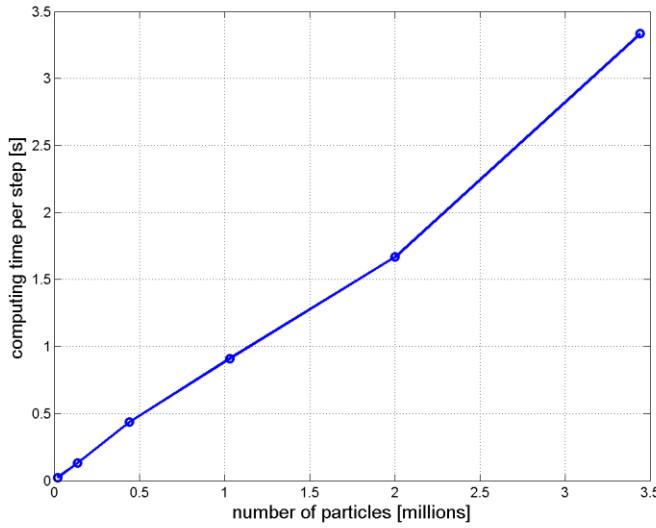


Figure 6.22: Computing time for one simulation step, plotted over the number of SPH particles in the simulation. The times are recorded for the simulations above on a NVIDIA GeForce 8800 GTX graphic card.

It can be seen that the implementation, described in section 5.2, scales linearly with the number of SPH particles. However, due to stability restrictions, the integration time step length has to be refined proportionally to the resolution (see Eq. (88)). The time step length, Δt , used in the above simulations are recorded in Table 6.5.

Table 6.5: Time step length, Δt , with respect to the number of particles.

number of particles:	17,576	132,651	438,979	1,030,301	2,000,376	3,442,951
time stepping, Δt :	$1.0e^{-6}$	$7.5e^{-7}$	$5.0e^{-7}$	$3.75e^{-7}$	$3.0e^{-7}$	$2.5e^{-7}$

Figure 6.23 illustrates the total computing time for the whole simulations dependent on the number of SPH particles.

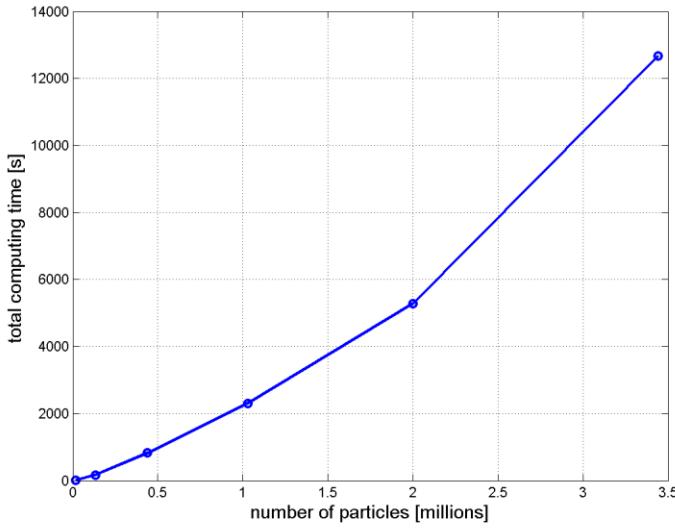


Figure 6.23: Total simulation time on a NVIDIA GeForce 8800 GTX graphic card, recorded for the simulations above.

A work precision diagram, that relates simulation time to accuracy, can be obtained combining Figure 6.21 and Figure 6.23.

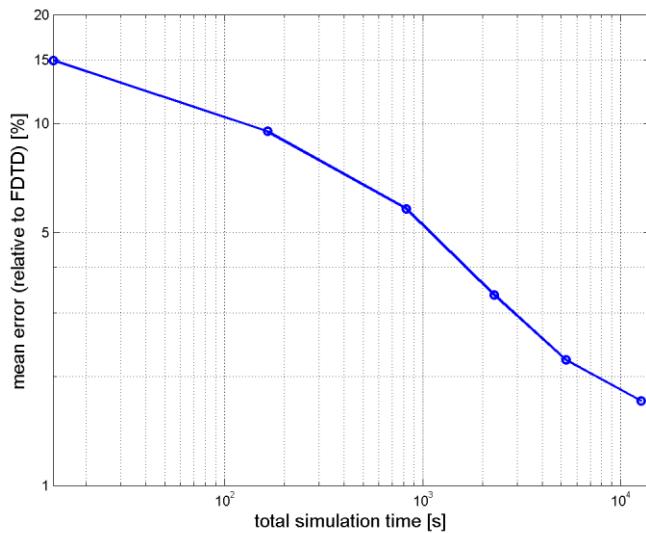


Figure 6.24: Work-precision diagram for the acoustic SPH simulations above. The achieved precision is plotted over the total simulation time on a NVIDIA GeForce 8800 GTX graphic card.

The number of particles increases cubically and the accuracy increases quadratically with resolution. Thus, the total simulation time increases slightly more than quadratically with the achieved accuracy. For this reason, it is essential to choose a resolution which results in both low simulation time and good accuracy.

Chapter 7

Conclusion and Future Work

This study described the current state of numerical simulation in acoustical engineering and pointed out the lack of a general solution to the acoustic modeling problem when dealing with complex environments or nonlinear propagation media. To address these shortcomings, two different particle methods have been investigated in order to assess their potential use in acoustic simulations. A lumped mass model was developed which displayed nonlinear propagation patterns in agreement with known solutions. Due to the current limitation to one-dimensional modeling, another Lagrangian particle method called Smoothed Particle Hydrodynamics (SPH) was considered, leading to more promising results. The numerical experiments conducted in this study indicate that it is possible to simulate sound wave propagation using SPH. Initial results are in agreement with the solution of the linear wave equation calculated using a FDTD scheme. Small shape deviations are the result of slightly anisotropic wave propagation speed which has been analyzed in a set of one-dimensional simulations. Different boundary formulations available in literature, along with a newly developed one, have been used to model rigid walls in a three-dimensional tube experiment. Using dynamic boundary particles and the mirror particle approach, the excitation of pressure waves through moving boundaries has been simulated in good agreement with the analytical solution and, moreover, sound wave reflection at rigid walls was reproduced accurately.

Furthermore, it has been shown that three-dimensional problems with high resolution can be simulated in reasonable time by drawing on the developed GPU-based SPH implementation.

It has been shown that meshless Lagrangian methods have advantages over mesh-based methods making them attractive for acoustic simulations. In conclusion, the use of SPH can prove especially beneficial in cases where other methods have difficulties. Specifically, these cases include:

- aero-acoustical problems
- complex and changing domain topologies
- domains with multiple propagation media
- domains with high temperature or density gradients
- nonlinear acoustics and shock waves with fluid-structure interaction

However, to this end, deficient boundary formulations result in noisy simulations. For this reason SPH still far from being applicable in real world acoustic simulations. Moreover, results get corrupted as soon as the initial particle placement doesn't represent an equilibrium constellation. This leads to the fact that only simple grid structures can be used to this point. Furthermore, it takes some experience to choose all SPH parameters correctly. While the SPH algorithm is generally very stable, a bad choice of parameters can either lead to instable simulations or inaccurate results.

Nonetheless, this work has demonstrated the general capability to model acoustic problems using SPH. Specific contributions of the author include the following:

- Analysis of meshless Lagrangian methods with focus on applicability in acoustical engineering

- Lumped mass model of one-dimensional nonlinear sound propagation
- Implementation of SPH on the CPU using Matlab and on the GPU using CUDA
- Method to model fluid structure interaction through spherical decomposition
- Analysis of the impact of smoothing length on wave speed
- Analysis of sound excitation due to moving boundaries
- Modified equation of state for ideal gas that removes stability issues
- New simplified first principal boundary conditions for the use in hydrodynamic SPH simulations
- Work-precision diagram for an acoustic SPH simulation

Ongoing work will primarily seek to find a solution to problems associated with currently available boundary formulations. In this context it has been reported that a corrective SPH formulation, called CSPH, reduces disturbances near boundaries drastically [26]. Clearly, further work needs to be done in order to validate the method for nonlinear wave propagation as in, for example, blast simulations. Another interesting application of SPH to sound propagation through inhomogeneous media remains to be addressed as well.

Although physically meaningful simulations are complicated and require better suited boundary formulations, SPH has the potential to become a valuable tool in acoustical engineering.

References

- [1] Auld, B., *Acoustic fields and waves in solids*: Krieger Pub Co, 1990.
- [2] Suzuki, H., Omoto, A., and Fujiwara, K., Treatment of boundary conditions by finite difference time domain method. *Acoustical Science and Technology*, 28(1), 2007. p. 16-26.
- [3] Hickling, R., Feldmaier, D., Chen, F., and Morel, J., Cavity resonances in engine combustion chambers and some applications. *The Journal of the Acoustical Society of America*, 73, 1983. p. 1170.
- [4] Bettess, P. and Zienkiewicz, O., Diffraction and refraction of surface waves using finite and infinite elements. *International Journal for Numerical Methods in Engineering*, 11(8), 1977.
- [5] Gaul, L., Kögl, M., and Wagner, M., *Boundary element methods for engineers and scientists: An introductory course with advanced topics*: Springer, 2003.
- [6] Craggs, A., A finite element method for the free vibration of air in ducts and rooms with absorbing walls. *Journal of Sound and Vibration*, 173(4), 1994. p. 568.
- [7] Ciskowski, R. and Brebbia, C., *Boundary element methods in acoustics*: Elsevier Applied Science, 1991.
- [8] Botteldooren, D., Finite-difference time-domain simulation of low-frequency room acoustic problems. *Journal of the Acoustical Society of America*, 98(6), 1995. p. 3302-3308.
- [9] Murphy, D., Kelloniemi, A., Mullen, J., and Shelley, S., Acoustic modeling using the digital waveguide mesh. *IEEE Signal Processing Magazine*, 24(2), 2007. p. 55.
- [10] Funkhouser, T., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., West, J., Pingali, G., Min, P., and Ngan, A., A beam tracing method for interactive architectural acoustics. *The Journal of the Acoustical Society of America*, 115, 2004. p. 739.
- [11] Batchelor, G., *An introduction to fluid dynamics*: Cambridge Univ. Press, 2000.
- [12] Landau, L. and Lifshitz, E., Fluid Mechanics, Vol. 6. *Course of Theoretical Physics*, 1987.
- [13] Danforth, A. and Long, L., Nonlinear acoustic simulations using direct simulation Monte Carlo. *The Journal of the Acoustical Society of America*, 116, 2004. p. 1948-1955.
- [14] Yokokawa, M., Uno, A., Ishihara, T., and Kaneda, Y. 16.4-Tflops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator. 2002 IEEE Computer Society Press Los Alamitos, CA, USA.
- [15] Howe, M., *Acoustics of fluid-structure interactions*: Cambridge University Press, 1998.
- [16] Lighthill, M., On sound generated aerodynamically. I. General theory. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1952. p. 564-587.
- [17] Verlet, L., Computer experiments on classical fluids. II. Equilibrium correlation functions. *Phys. Rev*, 165(1), 1968.

- [18] Shatkay, H., The Fourier transform-A primer. *Brown University, Providence, RI*, 1995.
- [19] Beranek, L.L., *Acoustics*, New York: Acoustical Society of America, 1996.
- [20] Zabusky, N. and Kruskal, M., Interaction of solitons in a collisionless plasma and the recurrence of initial states. *Physical Review Letters*, 15(6), 1965. p. 240-243.
- [21] Sugimoto, N., Acoustic solitary waves in a tunnel with an array of Helmholtz resonators. *Journal of the Acoustical Society of America*, 99(4), 1996. p. 1971-1976.
- [22] Dodd, R., Eilbeck, J., Gibbon, J., and Morris, H., *Soliton and nonlinear wave equations*. Academic Press, New York, 1982.
- [23] Lucy, L., A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(12), 1977. p. 1013-1024.
- [24] Gingold, R. and Monaghan, J., Smoothed particle hydrodynamics- Theory and application to non-spherical stars. *Royal Astronomical Society, Monthly Notices*, 181, 1977. p. 375-389.
- [25] Benz, W. Smooth particle hydrodynamics-a review. 1990.
- [26] Liu, G.R. and Liu, M.B., *Smoothed particle hydrodynamics: A meshfree particle method*, New Jersey: World Scientific, 2003.
- [27] Monaghan, J.J., Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30, 1992. p. 543-574.
- [28] Ellero, M., Smoothed Particle Dynamics Methods for the Simulation of Viscoelastic Fluids. 2004.
- [29] Monaghan, J. and Gingold, R., Shock simulation by the particle method SPH. *Journal of computational physics*, 52(2), 1983. p. 374-389.
- [30] Monaghan, J., Simulating free surface flows with SPH. *Journal of Computational Physics*, 110, 1994. p. 399-399.
- [31] Monaghan, J. and Kos, A., Solitary waves on a Cretan beach. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 125, 1999. p. 145.
- [32] Cleary, P., Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling*, 22(12), 1998. p. 981-993.
- [33] Cleary, P., Ha, J., Alguine, V., and Nguyen, T., Flow modelling in casting processes. *Applied Mathematical Modelling*, 26(2), 2002. p. 171-190.
- [34] Monaghan, J.J., An Introduction to SPH. *Computer Physics Communications*, 48(1), 1988. p. 89-96.
- [35] Cole, R.H., Underwater Explosions. *Princeton University Press*, 1948.
- [36] Iserles, A., *A first course in the numerical analysis of differential equations*: Cambridge University Press, 1996.
- [37] Courant, R., Friedrichs, K., and Lewy, H., On the partial difference equations of mathematical physics. 1959, CALIFORNIA UNIV LOS ANGELES.
- [38] Monaghan, J. and Lattanzio, J., A refined particle method for astrophysical problems. *Astronomy and Astrophysics (ISSN)*, 149(1), 1985. p. 135-143.
- [39] Morris, J., *Analysis of smoothed particle hydrodynamics with applications*: Monash University, 1996.
- [40] Monaghan, J.J., Smoothed Particle Hydrodynamics. *Reports on Progress in Physics*, 2005.

- [41] Libersky, L. and Petschek, A. Smooth particle hydrodynamics with strength of materials. 1990 Springer.
- [42] Biggs, M., Roy, A., and Hughes, C., First principles treatment of boundary conditions in SPH. *University of Edinburgh*, 2007.
- [43] Kajtar, J. and Monaghan, J., SPH simulations of swimming linked bodies. *Journal of Computational Physics*, 227(19), 2008. p. 8568-8587.
- [44] Heyn, T., Tasora, A., Anitescu, M., and Negrut, D., A Parallel Algorithm for Solving Complex Multibody Problems with Stream Processors, in ASME-IDETC. 2009, ASME: San Diego, CA.
- [45] Wolfe, C.T. and Semwal, S.K., Acoustic Modeling of Reverberation using Smoothed Particle Hydrodynamics. *Wscg 2008, Full Papers*, 2008. p. 191-198, 214.
- [46] Fahy, F. and Rossing, T., Foundations of engineering acoustics. *The Journal of the Acoustical Society of America*, 111, 2002. p. 1142.
- [47] Monaghan, J., SPH compressible turbulence. *Journal reference: Mon. Not. Roy. Astron. Soc*, 335, 2002. p. 843.
- [48] Violeau, D. and Issa, R., Numerical modelling of complex turbulent free-surface flows with the SPH method: An overview. *International Journal for Numerical Methods in Fluids*, 53(2), 2007.
- [49] Liu, M., Liu, G., Zong, Z., and Lam, K., Computer simulation of high explosive explosion using smoothed particle hydrodynamics methodology. *Computers and Fluids*, 32(3), 2003. p. 305-322.
- [50] Johnson, G., Stryk, R., and Beissel, S., SPH for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4), 1996. p. 347-373.
- [51] Mazhar, H., GPU Collision Detection Using Spatial Subdivision with Applications in Contact Dynamics, in ASME-IDETC. 2009, ASME: San Diego, CA.
- [52] Jones, D. and Belton, D., *Smoothed Particle Hydrodynamics: Applications Within DSTO*: Defence science and technology organisation Victoria (Australia) Platform science lab, 2006.
- [53] NVIDIA Corporation, NVIDIA CUDA: Compute Unified Device Architecture, Programming Guide Version 2.3. 2009, NVIDIA Corporation: Santa Clara.
- [54] NVIDIA Corporation. TESLA C1060 Datasheet. 2008; Available from: http://www.nvidia.com/docs/IO/43395/NV_DS_Tesla_C1060_US_Jun08_FINAL_LowRes.pdf.
- [55] POVRAY, The Persistence Of Vision RAYtracer. 2006.
- [56] Hoffman, J., *Numerical methods for engineers and scientists*: CRC Press, 2001.
- [57] Monaghan, J., Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1), 1992. p. 543-574.
- [58] Myint-U, T. and Debnath, L., *Partial differential equations for scientists and engineers*: New York-Amsterdam-London: North Holland (Elsevier Science Publishing Co, 1987.