# Vergleich der hydrodynamischen Simulationsmethode MFM zu SPH und Mesh-basierenden Verfahren

**Lucas Kimmig**


München 2018

# Comparison of the hydrodynamical simulation method MFM to SPH and mesh-based methods

**Lucas Kimmig**


München 2018

# Comparison of the hydrodynamical simulation method MFM to SPH and mesh-based methods

**Lucas Kimmig**

Bachelorarbeit
an der USM
der Ludwig–Maximilians–Universität
München

vorgelegt von
Lucas Kimmig
aus Wiesbaden

München, den 19.07.2018

Erstgutachter: Dr. Benjamin Moster
Tag der mündlichen Prüfung: 26.07.2018

# Table of Contents

# List of Figures

# Abstract

This bachelor thesis presents a look at the most common simulation codes implemented in an astrophysical context. It provides a summary of these main codes, namely fixed/moving mesh, smoothed particle hydrodynamics and the newer Lagrangian methods of meshless finite mass and meshless finite volume as pioneered by Philip Hopkins. We will give an outline of these methods, their strengths and weaknesses and a broad description of their implementation. Additionally, we explicitly compare applications of SPH and MFM on a variety of typical hydrodynamic tests to test each codes capabilities in the most common applications for astrophysics. The goal of this thesis is to analyze the MFM method and observe its properties in a wide variety of situations.

# Abstract

Diese vorliegene Bachelorarbeit präsentiert die gängisten Simulationen von denen in einem Astrophysikalischen Kontext Gebrauch gemacht werden. Es handelt sich dabei um die Kategorien von fixed/moving mesh, smoothed particle hydrodynamics sowie die neueren Methodiken von Philip Hopkins, namentlich meshless finite mass und meshless finite volume. Wir werden eine grobe Skizze dieser Methodiken präsentieren, ihre Vor- und Nachteile sowie eine Beschreibung der Implementierung. Des Weiteren wird von einer SPH Version gebrauch gemacht, um explizit an den gängisten hydrodynamischen Tests die Fähigkeiten von MFM zu analysieren. Das Ziel dieser Arbeit ist dadurch einen genauen Einblick in MFM und dessen Kapazitäten zu liefern.

# Chapter 1

# Introduction

As our universe presents us with a plethora of fascinating objects and processes to explore, it seems natural that we have tried devising methods of explaining what lies before us, simplifying it into forms we can work with, analyze and understand. One such method is that of astrophysical simulation, a necessary procedure to analyze most astrophysical processes in a dynamic context as these by far surpass our life timescales. Additionally, the universe does not present us with a neat observation location and galaxies or stars in every stage of their life, so that we may exactly observe their evolution in time. The large-scale proceedings, such as structure formation or supernovae, often only have a few relevant contributors, for example to model supernova remnants the gravitational effects can be largely ignored. These simplifications, such as modeling the interstellar gas with an ideal gas, while 'only' being approximations, still can aid our understanding of physical procedures that in many cases cannot be solved analytically. They help us check the predictions of different models up to high accuracies. The use of fluid dynamics in an astrophysical context is therefore justifiable as a highly precise approximation of the processes involved in inter alia galaxy formation or supernovae, just as the use of LCAO is for the orbits of the atoms in helium.

A wide variety of different tools for such simulations have been devised throughout the years. For example, smoothed particle hydrodynamics ('SPH') has been around for 40 years now - it was presented by Monaghan and Gingold already in 1977 [16] - and therefore has gone through quite a lot of optimization, as is also the case for mesh-based methods. We, however, will focus on one of the more recent developments with the introduction of the meshless finite mass and volume methods ('MFM' and 'MFV' respectively) by Phil Hopkins in 2015 [8], and will compare these in their structure and theoretical foundation to the two aforementioned most prevalent main categories: SPH (such as GADGET-2 [24]) and mesh-based methods, of which the later can be moving (such as AREPO [25]) or fixed mesh (such as ATHENA [26]). Additionally, we will conduct a row of test simulations and compare the results from a modern SPH implementation with those of MFM. As the areas and tests explored with these methods vary widely, the advantages and disadvantages of each code can be more or less relevant depending on the problem at hand.

This work is structured as follows: First in chapter 2 we will provide a derivation of the

most relevant physical formulas and therefore the origins of the governing equations for our tests. Then we move on to the numerical methods and their implemented principles, providing a sketch of their differences in chapter 3 before moving on to the hydrodynamic test cases in chapter 4. There we discuss the implemented initial conditions as well as some physical background to the tests and any specific formulas needed. After presenting the simulation results of our test cases run with SPH and MFM in chapter 5, we finally conclude our thoughts on the new method MFM in light of its competitors in chapter 6.

# Chapter 2

# Hydrodynamics

## 2.1 Euler equations

As argued in the introduction 1, a plethora of astrophysical problems can be approximated by problems of fluid dynamics. In this context a fluid means a continuous medium, where even small volume elements (small relative to the total volume) of the fluid contain a large number of molecules. The most important equations are the Euler equations, which represent the continuity of mass, momentum conservation and energy conservation, as they govern the dynamical evolution of the system. We will follow the derivation of Landau and Lifshitz [13].

### 2.1.1 Continuity of mass

To derive the equation for the continuity of mass, we start with a reasonable requirement: that the total mass be conserved, and therefore also the total mass flow. The total mass is the density integrated over the volume, and the flow is defined to be positive if it leaves the volume. Then the flow of mass out of the volume through the surface $\mathrm{d}\mathbf{A}$ must equal the total mass loss (loss ergo the negative time derivative) in the volume dV itself

$$\oint_{\partial V} \rho \mathbf{v} \cdot \mathrm{d}\mathbf{A} = -\frac{\partial}{\partial t} \int_V \rho \, \mathrm{dV} \tag{2.1}$$

Using the divergence theorem, the surface integral on the left side is converted into a volume integral over the divergence of its arguments, and we assume the density to be smooth enough to swap the time derivative and the volume integral on the right hand side

$$\int_V \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) \, \mathrm{dV} = \int_V \left( - \frac{\partial}{\partial t} \rho \right) \mathrm{dV} \tag{2.2}$$

$$\int_V \left( \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) + \frac{\partial}{\partial t} \rho \right) \mathrm{dV} = 0 \tag{2.3}$$

As this must hold for any volume, it follows that the integrands must always equal 0, from which we can then derive the continuity equation as

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) = 0 \tag{2.4}$$

### 2.1.2 Momentum conservation

By starting from the total force that a fluid exerts on a volume it surrounds, and equating this with the force from acceleration (of the fluid within the volume), we derive the Euler equation of momentum conservation. As the pressure (p) is defined as the force per area, the total force is equal to the pressure over the total surface area of the fluid or, again using the divergence equation, equal to the gradient of the pressure over the volume element.

$$-\oint_{\partial V} p \, \mathrm{d}\mathbf{A} = -\int_V \boldsymbol{\nabla} p \, \mathrm{dV} \tag{2.5}$$

A minus appears due to the definition of the surface integral (force acting on the surface is noted with a minus sign, force acting out of the surface is noted with a plus sign). The gradient of the pressure therefore acts as a force per volume element, and is equal to the force per volume element given by the acceleration times the density

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\boldsymbol{\nabla} p \tag{2.6}$$

As the acceleration is the total time derivative of the velocity, but the velocity depends both on the location and the time, we must expand the velocity as such (using $\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v}$ ):

$$\mathrm{d}\mathbf{v} = \frac{\partial \mathbf{v}}{\partial t}\mathrm{d}t + (\mathrm{d}\mathbf{r} \cdot \boldsymbol{\nabla})\mathbf{v} = \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \boldsymbol{\nabla})\mathbf{v}\right)\mathrm{d}t \tag{2.7}$$

Plugging this into 2.6 we arrive at

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \boldsymbol{\nabla})\mathbf{v} = -\boldsymbol{\nabla} p \tag{2.8}$$

As there may be other external acceleration fields working on the fluid, there is an additional term $\rho\mathbf{g}$. Adding this to the right hand side of 2.8 and dividing everything by $\rho$ we get the Euler momentum conservation equation:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \boldsymbol{\nabla})\mathbf{v} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \mathbf{g} \tag{2.9}$$

### 2.1.3 Entropy conservation

There are multiple choices for the last of the five necessary equations to solve for the five variables that fully describe our fluid dynamics system $(\rho, p, v_x, v_y, v_z)$. One can choose

energy conservation or pressure conservation or, for special cases (like a Sedov blast), it can instead be useful to use the absolute time derivation of a different measure of entropy than the usual thermodynamical version (although we will be later employing an energy based scheme, it should be noted that there are other equally valid conservations laws involving entropy).

For the sake of completeness and as we will be using this definition of the entropy when comparing the plots for the Sod shock, we give a rough sketch of the entropy formula derivation here. We start with the definition for the heat capacity at constant volume for an ideal gas (where here E is the total internal energy, not to be confused with the total energy per volume used later for the energy derivation)

$$C_V \equiv T\frac{\partial S}{\partial T}\bigg|_V = \frac{\partial E}{\partial T}\bigg|_V \tag{2.10}$$

The total differential of the entropy is known from thermodynamics as

$$dS(T,V) = \frac{\partial S}{\partial T}\bigg|_V dT + \frac{\partial S}{\partial V}\bigg|_T dV = \frac{C_V}{T}\,dT + \frac{\partial p}{\partial T}\bigg|_V dV \tag{2.11}$$

where we have used a Maxwell relation for the second summand. Integrating over the temperature and volume, and using the ideal gas law to determine the pressure derivative, then gives us the entropy. Note that we can integrate separately as $C_V$ is independent of the volume for an ideal gas. This derivation therefore relies entirely on our assumption that interstellar gas can be approximated as an ideal gas.

$$S(T,V) = \int \frac{C_V}{T}\,dT + \int \frac{Nk_b}{V}\,dV = C_V\,ln(T) + Nk_b\,ln(V) = C_V\,ln\Big(TV^{\frac{Nk_b}{C_V}}\Big) \tag{2.12}$$

By observing that for an ideal gas we additionally have $T = pV/Nk_b$, $\gamma = C_P/C_V = 5/3$ and $C_V = \frac{3}{2}Nk_b$ then this becomes

$$S(T,V) = C_V\,ln\left(\frac{pVV^{\frac{2}{3}}}{Nk_b}\right) = C_V\,ln\Big(\frac{pV^\gamma}{m^\gamma}\Big) + C_V\,ln\Big(\frac{m^\gamma}{Nk_b}\Big) \tag{2.13}$$

As the total mass is conserved, $k_b$ is the Boltzmann constant and for our tests the number of particles is conserved (N=const.), the final term is just an additive constant and can be ignored. The relation between pressure, density and entropy then is

$$S(T,V) = C_V\,ln\Big(\frac{p}{\rho^\gamma}\Big) + constant \tag{2.14}$$

We can then use $S' \equiv \frac{p}{\rho^\gamma}$ as an entropy measure (as entropy is a quantity only relevant in comparison, an additive term is just a choice we can freely set to 0, and as the logarithm function is sufficiently smooth the bijectivity necessary between S and $S'$ is given). Therefore, we can deduce the evolution of the entropy (as entropy is a conserved quantity for

those tests where we would use it instead of energy conservation, so the total derivative is 0 there) via

$$\frac{\mathrm{d}S'}{\mathrm{d}t} = \frac{\partial S'}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})S' = 0 \tag{2.15}$$

The final 3 equations used then would be 2.4, 2.8 and 2.15 giving us 5 equations for 5 variables (3 from velocity, 1 each from density and pressure)

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\mathbf{v}) = 0 \tag{2.16}$$

$$\rho\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} = -\boldsymbol{\nabla}p \tag{2.17}$$

$$\frac{\partial}{\partial t}\Big(\frac{p}{\rho^\gamma}\Big) + (\mathbf{v}\cdot\boldsymbol{\nabla})\frac{p}{\rho^\gamma} = 0 \tag{2.18}$$

An easy way to quickly understand that this is indeed a valid measure for entropy is by observing that isentropic processes, i.e. those where the entropy stays constant, are defined in a p-V-diagram by the curve $pV^\gamma = const$, which (for constant total mass, the case in all our tests) is equal to the condition $p/\rho^\gamma = const$. Then, S' is constant exactly when S is constant (this is obviously not a legitimate proof, but a quick way to rationalize the result of the more rigid derivation above).

### 2.1.4 Energy conservation

We then turn to energy conservation, where for the derivation we start out from the total energy per volume (from here on E) and look at how it varies with time

$$E \equiv \tfrac{1}{2}\rho v^2 + \rho e \tag{2.19}$$

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial t}\left(\tfrac{1}{2}\rho v^2 + \rho e\right) \tag{2.20}$$

where $e \equiv E_{internal}/mass$ is the specific internal energy, therefore the potential energy per volume term, and $\frac{1}{2}\rho v^2$ is the corresponding kinetic energy per volume term. This then is the starting point for the energy equation, just asking how it evolves in time.

As the actual derivation is relatively tedious, we will just reference the derivations in the appendix 6 here and give the final equation, representing the energy conservation equation:

$$\frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2 + \rho e) = -\boldsymbol{\nabla}\cdot\big[\rho\mathbf{v}(\tfrac{1}{2}v^2 + w)\big] \tag{2.21}$$

We now have 5 equations (1 each from mass and energy conservation, 3 from momentum) for our 5 main variables (energy or specific energy, density and velocity in 3 spatial directions). However, our equations for the energy conservation additionally depend on the heat functional $w$ while those for mass and momentum additionally depend on pressure. This means that, while we know the relation between density, energy and the heat functional (as energy and the heat functional are easily linked through thermodynamics), this

is not the case for pressure. We need a further dependency between our variables to solve this differential system, which can be found in the relation between pressure, density and energy.

## 2.2 Polytropic ideal gas

An ideal polytropic gas is a gas that, in addition to fulfilling the ideal gas law, also obeys $pV^a = constant$, with $a$ being the polytropic index, $p$ the pressure and $V$ the volume. When we are dealing with such a gas, it is sensible to simplify some of the above equations by deducing the relation between specific internal energy, adiabatic coefficient, pressure and density. For an ideal gas the internal energy is

$$E_{internal} = nC_V T \tag{2.22}$$

where n is the amount of substance of gas (in moles), $C_V$ the heat capacity at constant volume V and T the temperature. We then look at the ideal gas law

$$pV = nRT \tag{2.23}$$

where R is the ideal gas constant. Solving this for $nT$ and plugging it into the internal energy 2.22 we arrive at

$$E_{internal} = \frac{C_V}{R}\,pV = \frac{C_V}{C_P - C_V}\,pV = \frac{1}{\gamma - 1}\,pV \tag{2.24}$$

where we have used $\gamma = C_P/C_V$ the constant adiabatic coefficient (which is equal to 5/3 for an ideal gas), $C_P$ the heat capacity at constant pressure and the relation $R = C_P - C_V$. Then dividing both sides by the mass, we obtain the polytropic gas equation of state (as $\rho = m/V$ and e is the specific energy)

$$e(\gamma - 1) = \frac{p}{\rho} \tag{2.25}$$

This can for example be used to represent the energy conservation formula in terms of the total energy per volume E. Using $e = w - pV/mass = w - \frac{p}{\rho}$ we get

$$\rho(\tfrac{1}{2}v^2 + w) = E + \rho(w - e) = E + p \tag{2.26}$$

Plugging this and $E = \frac{1}{2}\rho v^2 + \rho e$ into 2.21 we arrive at

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot [(E + p)\,\mathbf{v}] = 0 \tag{2.27}$$

This then is the final of our equations for an ideal gas approximated by fluid dynamics that govern the evolution of our systems:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) = 0 \tag{2.28}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \boldsymbol{\nabla})\mathbf{v} = -\boldsymbol{\nabla} p \tag{2.29}$$

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot [(E + p)\,\mathbf{v}] = 0 \tag{2.30}$$

The attentive reader will notice that there are 5 equations, but 6 variables (total energy per volume $E$, pressure $p$, density $\rho$ and the 3 parts of velocity $\mathbf{v}$). The sixth implicit equation is the equation of state 2.25, which links the specific internal energy $e$ to pressure and density, thereby letting us represent $E$ dependent only on pressure, density and velocity (which then constitute our final 5 state variables, i.e. the variables that fully describe our systems state).

At this point it should be briefly mentioned when our equations are actually valid, i.e. what type of a system they assume. Our governing equations are called the Euler equations, and they assume our flow to be *inviscid* and *adiabatic*, meaning with viscosity and thermal conductivity approaching 0. They are a simplification of the Navier-Stokes equations, with the main difference being that Navier-Stokes can assume a dissipative system, while the Euler equations are *conservation* equations (as can be seen from our derivation of them: they represent conservation of mass, momentum and energy/entropy). They are accurate for a large amount of test problems, although this also means there are cases where they do not accurately evolve the system.

If we turn back to the condition of the processes being adiabatic, we can constrain the speed of sound to a simpler equation. As we are then dealing with an isentropic gas, namely one for which the polytropic index is the adiabatic coefficient gamma, $a = \gamma$, we get a simple relation between the speed of sound $c$ in the medium and density, pressure and the polytropic coefficient $\gamma$. We start with the isentropic condition $pV^\gamma = constant$ and divide both sides by $(mass)^\gamma$, which is constant

$$p\frac{V^\gamma}{m^\gamma} = \frac{p}{\rho^\gamma} = const \tag{2.31}$$

$$p = const * \rho^\gamma \tag{2.32}$$

$$c = \sqrt{\frac{\partial p}{\partial \rho}} = \sqrt{const * \gamma\, \rho^{\gamma-1}} = \sqrt{\frac{p}{\rho^\gamma}\gamma\, \rho^{\gamma-1}} \tag{2.33}$$

$$c = \sqrt{\frac{\gamma p}{\rho}} \tag{2.34}$$

The Mach number of a particle is then defined by the relation between its velocity and the local speed of sound

$$M = \frac{v}{c} \tag{2.35}$$

## 2.3 Surface discontinuities and shocks

For our considered test cases, we often deal with discontinuities of the fluid variables across an infinitely small border (for example all calculations of mesh based schemes require the solving of the equations across an effective face) - namely of Riemann problems further discussed in section 3.2. Here we will look at the specific implications for the hydrodynamical equations for a simple problem of a flat face with differing densities, velocities and pressures on either side. As for shocks, we tend to deal with high Mach numbers, and therefore we must treat the density as variable and as such the fluid as compressible.

To simplify the equations, we consider a coordinate system at the surface. The x-axis is chosen as the normal axis, so that the surface is equal to the y-z plane. Then the mass flux per surface element that is normal to the surface must be conserved (mass that passes over the boundary must appear on the other side), so that with $\rho_{1,2} \equiv$ density to the left/right of the surface, $v_{x1,x2} \equiv$ velocity normal to the surface on the left/right

$$\rho_1 v_{x1} = \rho_2 v_{x2} \tag{2.36}$$

The same must be true for the momentum flux per surface area. We start out by observing

$$\rho \frac{\partial}{\partial t} v_i = \frac{\partial}{\partial t}(\rho v_i) - v_i \frac{\partial \rho}{\partial t} \tag{2.37}$$

and use the equation of continuity 2.4 in the form $\frac{\partial \rho}{\partial t} = -\frac{\partial(\rho v_j)}{\partial x_j}$ to arrive at

$$\rho \frac{\partial}{\partial t} v_i = \frac{\partial}{\partial t}(\rho v_i) + v_i \frac{\partial(\rho v_j)}{\partial x_j} \tag{2.38}$$

If we then start out from 2.8 we get

$$\rho \frac{\partial}{\partial t} v_i = -\left( \frac{\partial p}{\partial x_i} + \rho v_j \frac{\partial v_i}{\partial x_j} \right) \tag{2.39}$$

$$\frac{\partial}{\partial t}(\rho v_i) = -\left( \frac{\partial p}{\partial x_i} + \rho v_j \frac{\partial v_i}{\partial x_j} + v_i \frac{\partial(\rho v_j)}{\partial x_j} \right) \tag{2.40}$$

$$\frac{\partial}{\partial t}(\rho v_i) = -\left( \frac{\partial p}{\partial x_i} + \frac{\partial(\rho v_j v_i)}{\partial x_j} \right) \tag{2.41}$$

$$\frac{\partial}{\partial t}(\rho v_i) = -\frac{\partial}{\partial x_j}\left( p\delta_{ij} + \rho v_j v_i \right) \tag{2.42}$$

As the normal to the surface is along the x-axis, the terms with index j are only non-zero for x, so that the condition of momentum flux conservation becomes

$$p_1 + \rho_1 \, v_{x1}^2 = p_2 + \rho_2 \, v_{x2}^2 \tag{2.43}$$

And for the y and z directions

$$\rho_1 \, v_{x1} \, v_{y1,z1} = \rho_2 \, v_{x2} \, v_{y2,z2} \tag{2.44}$$

Finally, we can also require the energy flux through the surface to be conserved. The flux can be read off from 2.27 as

$$(E_1 + p_1)\, v_{x1} = (E_2 + p_2)\, v_{x2} \tag{2.45}$$

$$\left(\tfrac{1}{2}v_{x1} + e_1 + \frac{p_1}{\rho_1}\right) \rho_1 v_{x1} = \left(\tfrac{1}{2}v_{x2} + e_2 + \frac{p_2}{\rho_2}\right) \rho_2 v_{x2} \tag{2.46}$$

$$\tfrac{1}{2}v_{x1} + e_1 + \frac{p_1}{\rho_1} = \tfrac{1}{2}v_{x2} + e_2 + \frac{p_2}{\rho} \tag{2.47}$$

where in the last step we have used the boundary condition for the mass flux 2.36.

As we now only need the normal velocity for the following, the subscript x is dropped. We will now derive the relationship between the density values before and after a shock. It is a bit tedious, but the simple result is well worth it and is a strong measure for the quality of our employed codes in the Sedov blast wave test. Following some of the basic ideas of the derivation given by Anderson [1], we then use 2.25 to receive the specific enthalpy $h = \frac{1}{m}(E + pV) = e + \frac{p}{\rho}$ in terms of either the pressure and density or in terms of the speed of sound in the medium c

$$\frac{p}{\rho} = (\gamma - 1)\, e = (\gamma - 1)\left(h - \frac{p}{\rho}\right) \tag{2.48}$$

$$\frac{p}{\rho} = \frac{\gamma - 1}{\gamma}\, h = \frac{c^2}{\gamma} \tag{2.49}$$

where we have used equation 2.34 in the final step. This is a general result that applies for any system of an ideal gas in an insentropic process.
Then coming back to a surface continutiy and dividing 2.43 by 2.36 we get

$$v_1 + \frac{p_1}{\rho_1 v_1} = v_2 + \frac{p_2}{\rho_2 v_2} \tag{2.50}$$

$$v_1 - v_2 = \frac{1}{\gamma}\left(\frac{c_2^2}{v_2} - \frac{c_1^2}{v_1}\right) \tag{2.51}$$

Now looking at the energy flux equation 2.47 and replacing e by h, we have

$$h_1 + \tfrac{1}{2}v_1^2 = h_2 + \tfrac{1}{2}v_2^2 \equiv h_0 \tag{2.52}$$

and using this in 2.49

$$c_{1,2}^2 = (\gamma - 1)\, h_{1,2} = (\gamma - 1)\left(h_0 - \tfrac{1}{2}v_{1,2}^2\right) \tag{2.53}$$

$$(\gamma - 1)\, h_0 = c_{1,2}^2 + \frac{\gamma - 1}{2}\, v_{1,2}^2 \tag{2.54}$$

Plugging this definition for the speed of sound in 2.51 we get

$$v_1 - v_2 = \frac{1}{\gamma}\left(\frac{(h_0 - \frac{1}{2}v_2^2)}{v_2} - \frac{(h_0 - \frac{1}{2}v_1^2)}{v_1}\right) \tag{2.55}$$

$$v_1 - v_2 = \frac{\gamma - 1}{\gamma}\left(\frac{h_0}{v_2} - \frac{h_0}{v_1} + \frac{1}{2}(v_1 - v_2)\right) \tag{2.56}$$

$$1 = \frac{\gamma - 1}{\gamma}\left(\frac{h_0}{v_1 v_2} + \frac{1}{2}\right) \tag{2.57}$$

$$\frac{\gamma + 1}{2}\frac{1}{h_0\,(\gamma - 1)} = \frac{1}{v_1 v_2} \tag{2.58}$$

where in the final step we have used $\frac{1}{v_2} - \frac{1}{v_1} = \frac{v_1 - v_2}{v_1 v_2}$ and then divided everything by $v_1 - v_2$ We notice

$$v_1^2 = M_1^2 c_1^2 = M_1^2(\gamma - 1)(h_0 - \tfrac{1}{2}v_1^2) \tag{2.59}$$

and therefore conclude

$$\frac{\rho_2}{\rho_1} = \frac{v_1}{v_2} = \frac{v_1^2}{v_1 v_2} = M_1^2(\gamma - 1)(h_0 - \tfrac{1}{2}v_1^2)\,\frac{\gamma + 1}{2h_0\,(\gamma - 1)} \tag{2.60}$$

$$\frac{\rho_2}{\rho_1} = \frac{M_1^2\,(\gamma + 1)h_1}{2\,(h_1 + \frac{1}{2}v_1^2)} = \frac{M_1^2\,(\gamma + 1)}{2 + \frac{v_1^2}{h_1}} = \frac{M_1^2\,(\gamma + 1)}{2 + (\gamma - 1)\,M_1^2} \tag{2.61}$$

$$\frac{\rho_2}{\rho_1} = \frac{\gamma + 1}{2M_1^{-2} + \gamma - 1} \tag{2.62}$$

where in line one we have used 2.58 and 2.59, and in line two 2.54. This then is the equation for the immediate post shock density. For most shocks we are dealing with supersonic proportions (as will be the case for our Sedov blast), so the Mach number is large enough that its inverse goes towards zero. Finally, we have then:

$$\frac{\rho_2}{\rho_1} = \frac{\gamma + 1}{\gamma - 1} \tag{2.63}$$

From this we can derive the pressure starting from 2.43 using $\rho_1 v_1^2 = \frac{\gamma p_1}{c_1^2} v_1^2 = \gamma p_1 M_1^2$ and the just derived density relation 2.63

$$p_2 = p_1 + \rho_1 v_1^2 - \rho_2 v_2^2 = p_1 + \rho_1 v_1^2\left(1 - \frac{v_2}{v_1}\right) = p_1\left(1 + \gamma M_1^2\left(1 - \frac{\rho_1}{\rho_2}\right)\right) \tag{2.64}$$

$$\frac{p_2}{p_1} = 1 + \gamma M_1^2\left(1 - \frac{2M_1^{-2} + \gamma - 1}{\gamma + 1}\right) \tag{2.65}$$

$$\frac{p_2}{p_1} = \frac{2\gamma M_1^2 - (\gamma - 1)}{\gamma + 1} \tag{2.66}$$

# Chapter 3

# Methods: SPH vs Grid vs MFM

In the following pages, the general methods of each simulation approach will be introduced and briefly explained, so that after discussing the test cases in a more general context in chapter 4 we can finally in chapter 5 compare the results of the test cases with some background to the implemented codes. We begin with Riemann problems, as these lie at the heart of the mesh-based and MFM codes, formulate their implementation in a Godunov scheme and then finally provide a summary of SPH, mesh and MFM codes and their most basic principles.

## 3.1 Introduction and Terminology

In anticipation of the following, let us look at the problem we are trying to tackle with these simulations. We start with a system that has some initial state. This will entail a distribution of the state variables, so it will have some distribution of density in its volume, for example. One area may have higher or lower density. How do we simulate this? By assigning more mass per volume in the area of higher density, would be the obvious answer. As physically we are dealing with something akin to a continuum (the smallest physical length scale is negligible) but practically we have a finite amount of resources to simulate our state, we must *discretize* our volume. We will then need to put some 'points' in the volume that are assigned a mass. This could be done with all our mass points having an equal mass, so that higher density would be simulated by having more of these *points* in one area. Or we could give some mass points a higher mass *value*, so that we have an even distribution of points throughout our volume but can still account for density fluctuations across it. Whatever the case may be, we have some finite amount of points in space that are assigned a mass. These will be referred to as *mass points*, or simply *points*. These then can also be assigned values for the remaining state variables, such as energy or entropy.

This is our system now, a finite amount of points in space that are given values for our state variables. How do we go about tracking the evolution of the system in time? The points will interact with each other, governed in their motion by our choice of aptly named governing equations 2.30. Without going into too much detail, we state in advance

that SPH tracks these points individually and calculates their motions based on the other points (and their values for the state variables) in the near vicinity, while mesh-based codes adopt an entirely different approach: They, in addition to the mass points, will have what we call *mesh-generating particles*, or *particles*, that are locations in space around which a mesh is generated, effectively splitting up our entire volume into smaller *cells* around their respective particle. The fundamental difference lies therein, that our calculation of system variables, so of our state vectors, for mesh-based codes will originate from the *particles*, which need not but can be identical to the mass points, whilst for SPH our state vectors are calculated at the *points* themselves. The reason we state this in advance is in the hopes of clearing up the nomenclature in the transition from discussion of SPH to the mesh-based codes.

## 3.2 Riemann problem

Fundamentally in MFM and mesh-based schemes, there will be an implementation of a Riemann solver. Therefore, we must first briefly look over what a Riemann solver actually is supposed to solve, while later explaining why it is necessary. This solver will approximate or exactly derive the solution to a Riemann problem, which is an initial conditions problem with constant starting values, a discontinuous point and a differential equation governing the evolution of the system.

The differential equation for a simple 1 dimensional example of such a problem would be (with $\rho_0(x)$ constant and $u$ a function of $\rho(x)$ )

$$\frac{\partial \rho}{\partial t} + \rho_0 \frac{\partial u}{\partial x} = 0 \tag{3.1}$$

with initial conditions as such

$$\rho(x,0) \equiv g(x) = \begin{cases} \rho_L & \text{if } x < 0 \\ \rho_R & \text{if } x \geq 0 \end{cases} \tag{3.2}$$

For a very simple case the relationship between the velocity field $u$ and the density field $\rho$ could be linear

$$u(\rho(x,t)) = a\rho(x,t) \tag{3.3}$$

with $a$ being a constant. Then putting this into equation 3.1, we would have a linear differential equation

$$\frac{\partial \rho(x,t)}{\partial t} + a\rho_0 \frac{\partial \rho(x,t)}{\partial x} = 0 \tag{3.4}$$

to which the solution would be:

$$\rho(x,t) = g(x - a\rho_0 t) = \begin{cases} \rho_L & \text{if } x < a\rho_0 t \\ \rho_R & \text{if } x \geq a\rho_0 t \end{cases} \tag{3.5}$$

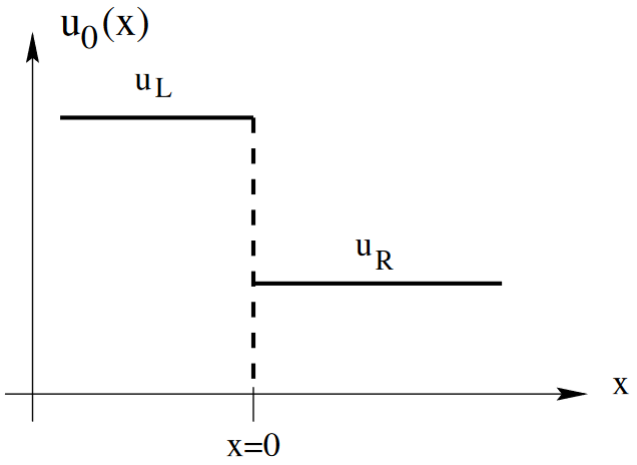



Figure 3.1: The initial conditions of our simple one-dimensional Riemann problem. We have a state variable $u$ changing discontinuously over a border, with constant values on both sides. The calculation of the flow between two particles at the effective faces will in essence be computed similarly to this simple setup.

Our solution $a\rho_0$ is the so-called characteristic (or eigenvalue) of our Riemann problem as explained well by Toro [28], which can be presented as a curve in an x-t plane. For our simple problem this would be a single line towards the top left or right, where on the left of the line we have density $\rho_L$ and on the right accordingly $\rho_R$. This solution describes a discontinuity moving to the right or left (depending on the sign of $a$) with a velocity of $a\rho_0$.

The presented differential equation is exactly the continuity equation presented in chapter 2. It arises as a governing equation for the hydrodynamics of the gas particles as it stems from mass conservation, which must be fulfilled for our tests. Before we go more in-depth into the functionality of mesh-based codes later in this chapter, let us pre-emptively state here that they create a mesh over our volume, i.e. form separate 'cells' around the aforementioned mesh-generating particles. As our system dynamically evolves over time, there are mass points and energy passing between the borders of these cells, so there are fluxes. To correctly simulate our system, we must then calculate these fluxes between the borders at each time step, functionally forming a row of Riemann problem.

The general solution to such a Riemann problem, with for example higher and lower density across a cell border, will be an interpolation of the higher and lower densities for some middle area between the particles before converging on either side to $\rho_L$ or $\rho_R$ respectively (for a neat visualization of this middle area of a differing value of for example density please refer to figure 3.3). We can make sense of this if we imagine, for example, a fluid container separated in the middle by a wall, filled higher on the left than the right. If we instantly remove the barrier and look at the evolution in slow motion, we would see the fluid directly at the border on the left slowly start falling as more fluid is pushed to the right-hand side through pressure equalization (the left side has higher pressure through gravity than the right side). A middle area would form with a fluid level lower than the

left but higher than the right, while the far left/right sides of the container would not yet be affected, so would still have their initial fluid levels. Given sufficient time, eventually the entire system would equilibrate at a level between the original left or right one.

Turning back to our hydrodynamic systems, we observe that we are normally dealing with more than one conserved property. This entails having more than one conservation law, more than one differential equation like 2.4 and therefore more than one characteristic of the system. If we briefly look how the above formalism can be expanded upon for multiple dimensions, we could consider a two-dimensional system described by an equation of motion of the form

$$\partial_t \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \partial_x \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = 0 \tag{3.6}$$

with initial conditions of

$$q_{1,2}(x, t = 0) = \begin{cases} q_{1l,2l} & \text{if } x < 0 \\ q_{1r,2r} & \text{if } x \geq 0 \end{cases} \tag{3.7}$$

to which the characteristics then would be $\lambda_\pm = \pm 1$ (so the eigenvalues of our matrix) with corresponding eigenvectors $e_\pm = \begin{pmatrix} 1 \\ \mp 1 \end{pmatrix}$. We then decompose the left and right states of our inital conditions into the eigenvectors with some factors $\begin{pmatrix} q_{1l} \\ q_{2l} \end{pmatrix} = a_+ e_+ + a_- e_-$ and $\begin{pmatrix} q_{1r} \\ q_{2r} \end{pmatrix} = b_+ e_+ + b_- e_-$. If we then solve this for $a_\pm$ and $b_\pm$, we get the final solution of the states of our three regions (left and right initial states and the newly arising interpolation state $a_+ e_+ + b_- e_-$ in the middle)

$$q_1(x, t) = \begin{cases} q_{1l} & \text{if } x \leq \lambda_- t = -t \\ q_{1m} = \frac{1}{2}\left[+(q_{1l} + q_{2l}) + (q_{2l} + q_{2r})\right] & \text{if } -t < x < t \\ q_{1r} & \text{if } x \geq \lambda_+ t = t \end{cases} \tag{3.8}$$

$$q_2(x, t) = \begin{cases} q_{2l} & \text{if } x \leq \lambda_- t = -t \\ q_{2m} = \frac{1}{2}\left[-(q_{1l} + q_{2l}) + (q_{2l} + q_{2r})\right] & \text{if } -t < x < t \\ q_{2r} & \text{if } x \geq \lambda_+ t = t \end{cases} \tag{3.9}$$

Here we then have two characteristics for each state variable that are moving linearly outwards from x=0 in time (one being $x_1 = t$, the other $x_2 = -t$), resulting in 3 total states.

This method can be and is expanded upon to account for discontinuities in any values across particle boundaries, which is then implemented in the form of the Riemann solvers. These solvers then calculate the evolution of the fluxes based upon the governing equations we derived in chapter 2 either exactly or, for sufficiently complicated fluxes, approximately. The adopted Riemann solver for MFM is chosen as the common approximate HLLC solver given by Toro [28].

## 3.3 Godunov scheme as a finite volume scheme

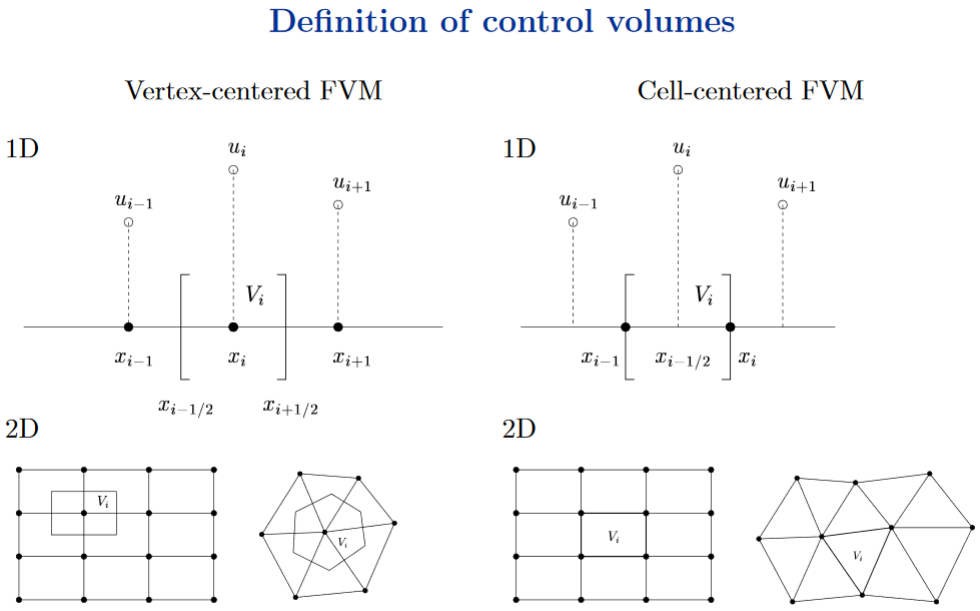



Figure 3.2: The different definitions of the control volumes. Here a vertex centered FVM will be employed

In principle the finite volume scheme is a specific way to partition the volume into a finite amount of points $\overline{q}_i$, which are then tracked over some amount of time. The word 'points' here means a point in space that is assigned a value for a specific state variable. As there are many options for how one could calculate the dynamic evolution of the system of state vectors, this presents only one such method in the family of finite volume schemes. For such schemes the conservation laws are applied in integral form, as this allows a wider range of systems, including discontinuities (Toro [28]). To receive the laws in integral form, the differential equations 2.30 are integrated over a control volume. Here a vertex-centered control volume is chosen (as described in figure 3.2), as this will have useful properties for the employed numerical methods later in this chapter, specifically for the mesh-based and MFM schemes.

To remember why such a scheme is useful, remember that we are facing the problem in our simulations of needing to track our finite amount of either mesh-generating particles or our mass points over time, as for mesh-based codes the particles and for SPH the mass points will contain our state variables. We update each of these points or particles after a timestep (which we will discuss more later, for now this just means a finite amount of time), so calculating with our governing equations the state a specific amount of volume around the point/particle will be in after a time $dt$. The application then of the above

concept in our codes should be obvious.

### 3.3.1 Principle of the Godunov scheme

We follow the implementation of the principle of the Godunov scheme by looking at its application on the fundamental form of the continuity equation 2.4 for one spatial dimension. The method works analogous for three dimensions and the other governing equations (albeit being more complicated there). For this example $q$ is our state variable in one spatial dimension dependent on space $x$ and time $t$, while $f$ is our field of fluxes dependent on $q$.

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{3.10}$$

By subtracting the second summand to the right-hand side and integrating over a control volume element $dx$ we arrive at

$$\frac{\partial}{\partial t} \int_{x_{i-1/2}}^{x_{i+1/2}} q \, dx = -(f(q(t, x_{i+1/2})) - f(q(t, x_{i-1/2}))) \tag{3.11}$$

We then define the spacing between the half-way borders of two points as $\Delta x \equiv x_{i+1/2} - x_{i-1/2}$ so that we arrive at a definition for the averaged value of our state vector of point/particle $i$ at timestep $n$ defined by the state variable integrated over the volume between the borders $x_{i\pm1/2}$ divided by $\Delta x$

$$\overline{q}_i^n \equiv \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(t^n, x) \, dx \tag{3.12}$$

By then integrating equation 3.11 over time we finally arrive at an exact update formula for each timestep:

$$\overline{q}_i^{n+1} = \overline{q}_i^n - \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \big(f(q(t, x_{i+1/2})) - f(q(t, x_{i-1/2}))\big) \, dt \tag{3.13}$$

If we can solve this integral, then by knowing the starting conditions at timestep $n = 0$, we can apply this formula iteratively to get the full evolution of our system in time. This is also evident by remembering that equation 3.10 represents our mass conservation equation 2.4 where $q = \rho$ and $f(\rho) = \rho v$, so that 3.13 represents a different form of the Euler equations governing the evolution of our system.

The key step in Godunov's scheme lies therein, that to avoid the complexities arising from trying to solve the exact time integral in each step

$$\int_{t^n}^{t^{n+1}} f(q(t, x_{i\pm1/2})) \, dt \tag{3.14}$$

said integral is *approximated* (originally Godunov used an Euler upwind scheme, which ensures accuracy up to the first order). This allows an analysis of the eigenstructure of the

system via the characteristics of our Riemann problem. How does this relate back to the Riemann problem? If we look at what we are integrating in equation 3.14 more closely, we notice that these are the fluxes at the borders $x_{i\pm1/2}$ between the points, and this is exactly what a Riemann solver can provide the solution to: the evolution, and therefore the fluxes, of a system with a discontinuous point between two states governed by a set of differential equations. This analysis then also lets us identify the upwind direction of the fluxes, which is later important for the numerical stability of the code (see [25] for more details). The important element to note here is that we have applied the Euler equations and evolved our system of state variables in time, arriving finally at a time integral we need to solve to get the next value for our state variable, which we will now approximate.

To achieve a second order accurate integration, we follow Springel [25] and arrive at

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = -\sum_j A_{ij} \cdot F_{ij} \tag{3.15}$$

$$\overline{q}_i^{n+1} = \overline{q}_i^n - \Delta t_i \sum_j A_{ij} \cdot \tilde{F}_{ij}^{n+1/2} \tag{3.16}$$

with $\tilde{F}_{ij}$ being an appropriate approximation for the fluxes and $\overline{q}_i^{n+1}$ is the volume averaged state variable over the cell $i$ at timestep *n+1*, while $A_{ij}$ describes the oriented area of the face between cells $i$ and $j$ (for derivation and more specifics see Colella 1990 [3], Hopkins 15 [8] or Stone et al. 2008 [26]). For now, this is merely a possible Godunov scheme to approximate the evolution of our system of state variables $q_i$ according to the initial differential equation 3.10 via discretization into multiple volume elements; however, it turns out to also be the form of the meshless equations of motion derived for MFM 3.6 and mesh-based codes 3.5 later in this chapter.

For sake of completeness we mention that MFM, like AREPO and other well-known codes, uses Godunov's method via the MUSCL-Hancock scheme. For more info on this specific implementation we would suggest the paper by van Leer [29] or the book by Toro [27].

### 3.3.2 The timescale

Now that we have the basic method and the specific approximation and scheme used, we then must choose a timescale for our particles to be updated. As we get a time by dividing a length through a velocity, it seems natural to choose the kernel length (which we will define in the next section) and signal velocity (the velocity that information is traveling at in our system, so how fast a particle will be feeling the effects of another particles motion) for our measure of time, arriving at

$$\Delta t_i = \frac{h_i}{|v_{sig,i}|} \tag{3.17}$$

$$v_{sig,i} = MAX_j \left( c_{s,i} + c_{s,j} - MIN \left( 0, \frac{(\mathbf{v}_i - \mathbf{v}_j)(\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|} \right) \right) \tag{3.18}$$

where we make sure to choose the maximum signal speed over all neighbors (the sum over $j$). Furthermore, MFM incorporates a limiter to prevent overly stark contrasts between two neighbors in terms of their timesteps (it would be disadvantageous if one particle is significantly slower in updating than its direct neighbor with which it shares fluxes). This then is the timestep chosen for MFM, which in addition to the limiter also incorporates a method from Springel [25] to ensure synchronized update of fluxes of conserved quantities across particle borders.

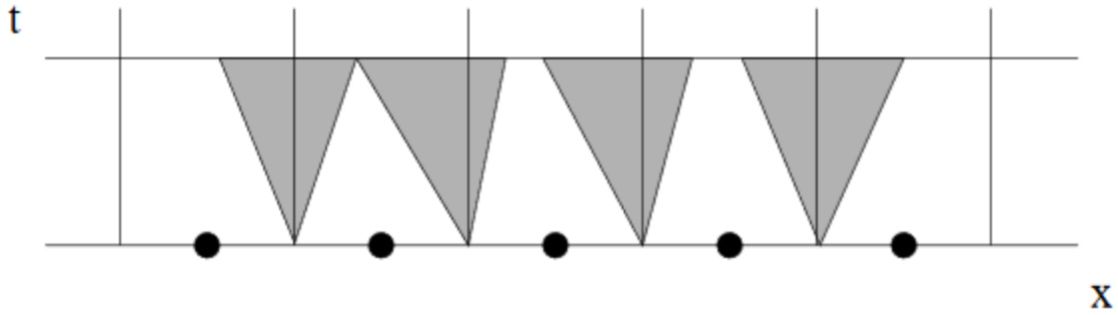



Figure 3.3: The limitations of the timesteps provided by the solutions of the Riemann problems in a Godunov finite volume scheme

The underlying principle of the choice of timestep and Godunov's method can be understood from figure 3.3: We solve Riemann problems at the effective faces of the particles (so at the borders of the cells present in mesh-based codes; for how this is implemented in SPH refer to the subsection 3.4.2), which lead to characteristics of the respective cells that change the respective cell's state variables and propagate inside the cell volume with a specific speed. This is accurate so long as these changing variables don't overlap with the characteristics of other cell pairs. If they overlapped we are solving two systems that are influencing each other as independent, which would obviously be faulty. Therefore, we must limit the timestep size, such that the solutions/information of the updates of the cell state do not reach the solutions of other cell pairs. Finally then, the solution for the updated cell's state variables is the average over the value of the characteristics from the side interfaces $i \pm 1/2$ and the untouched middle region (so the states initial state variables). For a more in-depth explanation, we refer to the courses by Dullemond [5].

## 3.4 SPH Scheme

SPH fundamentally represents an approximation of complex fluid dynamics by using a set of mass points, functionally discretizing the continuum. These mass points are assigned qualities such as energy and velocity and can interact with other points. Those it can interact with are called neighboring points, or their 'neighbors'. How these neighbors are chosen and how they interact with each other will then form the key factor in determining the functionality of our SPH code. Most implementations follow the general rule of

constructing a circle (for two dimensions) or a sphere (for three dimensions) around the points, and any other point falling into this range is then a neighbor and can interact with it. The exact weighting function for constructing this volume around the point will have an associated characteristic length, or smoothing length often denoted as $h$.

A key difference is that the points' associated volumes can *overlap* a multitude of times, meaning that (in theory) there is no limiting factor on how many nearby points influence each other (although nearly all common schemes do cap this number due to processing costs) and they can even have other particles between them and still interact, whereas for mesh-based methods the particles that influence each other share a stark border between them upon which the fluxes are calculated. If one were to visualize this, we can imagine three points that are sitting in a line. Now for SPH if they are sufficiently close they will all influence each other, meaning even the first and third share an interaction. For mesh-based methods however, the mesh borders of the first and third probably will not touch, which means they will not share a particle border and therefore don't share fluxes.

Depending on how the radius around the points then is constructed, we have different methods (as this radius determines which particles interact or not, it plays a fundamental role in our systems evolution). For a constant *radius* for example, we would need to count all the other mass points that are within this rigid volume (so all neighbors), whose number can change. This would be the method implemented by for example Springel and Hernquist [23]. Here however, we will be keeping the *number* of neighbors constant, leading to a *varying* characteristic length scale $h$ for each point (as implemented in the most common SPH codes such as Springels Gadget [24]). We also try to avoid constraining the number of neighbors to an actual discrete value, as this increases computation costs significantly and can lead to high fluctuations in h. If we imagine for example a strong clustering with all remaining particles being really far away, we might require a point in the clustered group to include one of the far out points to reach its required neighbor number, leading to a much larger h than is actually necessary to determine the evolution of the system (the far out particles' impact is negligible, so we just neglect them entirely to avoid wasting large amounts of resources). Based on this we choose the constraint with an allowed error of plus or minus up to one point, depending on the test at hand.

Finally then, we must evolve the system according to our discretized Euler equations. As mentioned in the chapter introduction, we will solve the equations at the location of the points themselves and they will depend on the neighbors' properties. These are then locally integrated. However, as we are dealing with points, in addition to using the integral form instead of the differential form so that we can capture the points adequately, we must also employ a distance-based weighting function that estimates the values of the neighbors at a specific point. This function is referred to as a kernel function, which is discussed in more depth in the following subsection.

The advantages of this meshless method are evident: The energy, linear and angular momentum, mass and entropy are all conserved (as the particles are discrete and their total number conserved), and due to a lack of a presupposed coordinate system (as is the case for mesh/grid-based methods) SPH is fully Galilean invariant (while this has also been achieved by for example Springel for moving-mesh codes [25], its implementation there

was significantly more difficult and therefore took longer). In addition, as the resolution is dependent on the number of surrounding points, an amalgam of such points will naturally have a higher resolution, thereby automatically following the flow of mass (more mass points in one area leads to a higher resolution of that area). Finally, as the scheme incorporates a finite amount N of points being followed individually that, unlike with most mesh-based codes, represent the *mass* points themselves, an integration of self-gravity follows naturally through a cosmological N-body code.

However, the full Lagrangian nature of the method has some drawbacks as well, namely issues with shock resolution, discontinuities and fluid mixing instabilities. The fact that SPH fares worse with contact discontinuities in comparison to mesh-based codes, which use a Riemann solver specifically designed to capture these well, is to be expected. The reason they struggle with shocks is that as the mass points in the shock are moving at high velocities, they are experiencing much shorter timesteps, while the points right outside are slow-moving and in a region of low density, so they therefore experience a lower resolution and longer timesteps. This effectively smooths out the sharp discontinuities experienced in a typical shock. The emergence of the fluid mixing instability, on the other hand, is discussed further in the fluid mixing tests 4.3.

To combat these problems, SPH in most cases is implemented with artificial viscosity and diffusion terms that facilitate mixing and shock resolution. These terms have been improved over the years and are now quite sophisticated, managing to apply finite viscosity and diffusivity only in the area of shocks, but not for for example in the presence of a shear velocity (if they still applied in the latter cases they would instantly rip appart structures such as the Gresho vortex or the Keplerian disk). The artificial viscosity terms, for example, provide zero viscosity *unless* the points are approaching each other in space. In this case it acts as a resistive pressure by converting the kinetic energy of the fast-moving points into thermal energy, mimicking the shock in advance (this can also be achieved by wake-up switches, which 'wake up' the slower points when the shock is arriving in the vicinity). However, as is their name, these switches are artificial and often must be chosen specifically to fit individual tests. They also represent a typical source for errors (as we are adding possible errors with the numerical implementation itself). Furthermore, SPH is only accurate to an order of 0, i.e. has errors even in the linear domain, namely the so-called E0 error.

It should be noted here that this and the following paragraphs represent just a broad summary of the most basic principles; we would direct the reader to Monaghan and Gingold's original SPH paper [16] for an excellent explanation of the fundamentals, to the more modern review also by Monaghan [19] for a more in-depth look at SPH and its evolution up to 2005, to the book by Violeau [30] for a full theoretical backdrop on fluid mechanics and SPH in its entirety or for a simplified overview to the github wiki pages [7].,

### 3.4.1 Formulation of SPH

As most of the relevant system properties can be derived given the density of the gas, the core of SPH relies on estimating the density through a so-called kernel-summation. The

kernels $W(\mathbf{r}, \mathrm{h})$ are required to be symmetrical and sufficiently smooth (see Monaghan [18] for all necessary kernel properties), and have the aforementioned characteristic length $h$. The choice of kernel can vary and will lead to different results. Then, any field $F(\mathbf{r})$ is smoothed through a convolution with said kernel.

$$F_s(\mathbf{r}) = \int F(\mathbf{r}) W(\mathbf{r} - \mathbf{r}', \mathrm{h}) \mathrm{d}\mathbf{r}' \tag{3.19}$$

This helps us determine an estimated density for a finite set of points (for a sufficiently dense sample the integration can be replaced with a sum):

$$\rho_s(\mathbf{r}) \simeq \sum_i \rho_i W(\mathbf{r} - \mathbf{r_i}, h) \frac{m_i}{\rho_i} \tag{3.20}$$

where $\frac{m_i}{\rho_i}$ is the volume element of each point. The kernels characteristic length should naturally be further apart than the spacing $d$ of the mass points $h \geq d$ which leads to a minimum of 33 neighbors in 3D (see Springels derivation for this in [25]), which imposes a stark bottom border on the number of neighbors. As in general the spacing of the points varies over the space (for example, if we choose equal mass for all points then the spacing often varies already for the initial conditions), it is elegant to adopt a space-dependent kernel length $h = h(\mathbf{r}, t)$.

As the kernel drops to zero at a certain distance for all commonly used kernels, for example with the cubic spline at $r = 2h$ (see the kernel subsection), one can restrict the sum over i to the particles within this radius, namely the neighbors $N_{NGB}$. The number of neighbors is kept nearly constant throughout, and the computational cost therefore (as we are calculating the updated state variables for $N$ points each based on $N_{NGB}$ other points) is of order $\mathcal{O}(N_{NGB}N)$. This is the reason for the speed and computational efficiency of SPH codes, as we prevent being of order $N^2$ while still retaining high accuracy. One could in theory also consider a kernel that does not go to zero and would therefore account for all particles in the volume, like for example a Guassian kernel. This does lead to an even higher accuracy, however is of order $N^2$ and therefore not worth the increase in computational costs.

This general approximation of the density as in equation 3.20 is then used in the Euler equations 2.30 and the equation of state 2.25 to determine the other state variables. This is done by following different schemes. A 'density-energy' scheme will evolve density and internal energy explicitly and then determine for example the pressure from the equation of state (and velocity then from momentum etc). These then are the approximated equations of motion, which avoid the difficulties with an exact time integral, as the fields are all smoothed (and therefore easier to integrate) and for dense enough regions even entirely replaced by a sum over discrete values.

Our particular implementation of SPH is of form of a 'pressure-energy' scheme following the PSPH implementation as presented in Gizmo in appendix F2 [8]. As we are only using what Hopkins calls PSPH and not TSPH we will henceforth be refering to PSPH only as SPH. Depending on the test we use either a cubic or quintic spline with an according number of neighbors (see section 3.6.2 for more details). An alternative to the pressure-energy

formulation would be the pressure-entropy implementation (where we follow the entropy instead of the energy as discussed in section 2.1.3), which also appears in more general forms with the pressure being defined over our monotonic function of the thermodynamic entropy $S'$ (one can then construct a condition for which such entropy formulations are identical to the energy formulations, which won't be presented here).

$$p_i = S_i'(S)\rho_i^\gamma \tag{3.21}$$

As the E0 error has been referenced as an inherent disadvantage for SPH, we briefly show where it comes from, following the derivation presented by Read, Hayfield and Agertz [21]. For this we look at our velocity and assume it is smooth and can be Taylor expanded up to the second order of our kernel length to get

$$\mathbf{v}_j \cong \mathbf{v}_i + h\left(\frac{\mathbf{r}_{ij}}{h}\cdot\nabla_i\right)\mathbf{v}_i + \mathcal{O}(h^2) \tag{3.22}$$

Plugging this and the respective Taylor expansion for the pressure into the momentum equations 2.8, we arrive at

$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} \cong -\frac{p_i}{h\rho_i}\mathbf{E}_{0,i} - \frac{(V_i\nabla_i)p_i}{\rho_i} + \mathcal{O}(h) \tag{3.23}$$

with $V_i$ the volume of particle $i$ and $\mathbf{E}_{0,i}$ being our dimensionless error vector. This is said unavoidable E0 error in the momentum equations for all SPH implementations.

### 3.4.2 Godunov SPH

As mentioned previously in chapter 3.4, the artificial viscosity terms in SPH represent a typical source for errors, as do shocks. There have been different ideas to resolve these, one of the more prominent ones being the implementation of the Godunov finite volume method as presented in section 3.3 in SPH by Inutsuka [11]. It has been shown (see for example Molteni 2003 [15]) that the aforementioned common error sources are avoidable by implementing a Riemann solver between the points (so if they overlap, instead of only using the weighting function in the integral, solving a full Riemann problem to determine the acting forces), quite like the Riemann solvers used in mesh-based methods. This is due to the fact that the necessity for artificial diffusion terms arises specifically due to mixing errors, which (like shocks) can be resolved well via approximation to Riemann problems. This is commonly referred to as *Godunov SPH*. However, as with 'regular' SPH the E0 errors remain, and whilst the Riemann solver helps alleviate the named common errors, it also leads to new ones appearing, as we then also need to implement a slope limiter, which itself is a common source of errors for mesh-based codes.

In anticipation of what is to come, we note that this idea (of using the Godunov finite volume method)) has been broadened and tested in a multitude of ways, as it brings great potential for a code that can resolve both gravity and mixing well, creating something of a mixture between mesh and meshless codes. For example, a possibility to implement the

general idea of a Godunov scheme in a meshless finite-volume method was constructed by Lanson and Vila (2008 I [14]), which ends up forming the closest comparative method to MFM and presents a method much closer to moving-mesh methods in nature than to SPH (as the fluxes are determined over 'effective' particle faces, see chapter 3.6).

## 3.5 Fixed mesh and moving meshes

We will now turn to mesh-based methods, which can be broadly summarized into the two main categories of fixed- and moving-mesh. For these methods, as opposed to SPH, the particles now form the basis for creating a grid, or mesh over the volume. The mesh-creating particles can therefore differ from the mass points (which is always the case for fixed-mesh methods and can be the case for moving meshes), although it can be of use to keep them moving identically as in SPH. Once we have constructed our grid, which normally takes up a large part of the overall computational costs, it can either be *moving* or it could be *fixed* in our volume, depending on whether or not the mesh-generating particles move. Furthermore, the mesh itself may deform and be refined dynamically *around* the particles, meaning even with fixed particle positions we could have the borders of the single cells change over time. This would be an example of an adaptive mesh refinement, or 'AMR' code. These in principle construct a rough basic mesh and then add finer meshes in areas of interest, therefore adaptively refining the resolution. On the other hand, the mesh could be static in form around the particles. These codes with a fixed mesh around the particle must only create the mesh once while AMR codes require constant 'maintenance', but in return will need to equally resolve the *entire* volume, even the areas that end up having barely any mass to none, as we do not know in advance where these areas will be.

The different types all offer advantages and disadvantages. Fixed meshes without border deforming and refining often provide lower computational costs and can provide very high resolutions (for example if the problems evolution is known ahead of time one can construct a mesh with higher resolution in the areas of higher density) but can struggle if the systems solution is not known, and especially with highly dynamic systems. Moving mesh codes with adaptive refinement on the other hand do not require the solution to be known in advance and can deal with a wide array of problems, although it can run into trouble when particle borders deform too much and create bad cells.

There are different ways of constructing a mesh around the particles, for example one could implement a Voronoi mesh as in figure 3.4 which is the method utilized in AREPO [25], or one could use coordinate transformations as implemented by Pen [20] or a Delaunay tessellation as used by for example Xu [32]. If we turn to the specific case of a voronoi mesh as a representative for the possible meshes, we can tell from the image that each particle has a clearly defined domain, so there are no shared masses between particles and therefore no weighting functions. These cells touch other cells along a line in 2, or a surface in 3 dimensions. This border is referred to as the 'face' of the cell. The fluxes between two particles then are evaluated on these faces between their volumes. Herein lies the crucial core of mesh-based codes, and their necessity for a Riemann solver, as in each step there

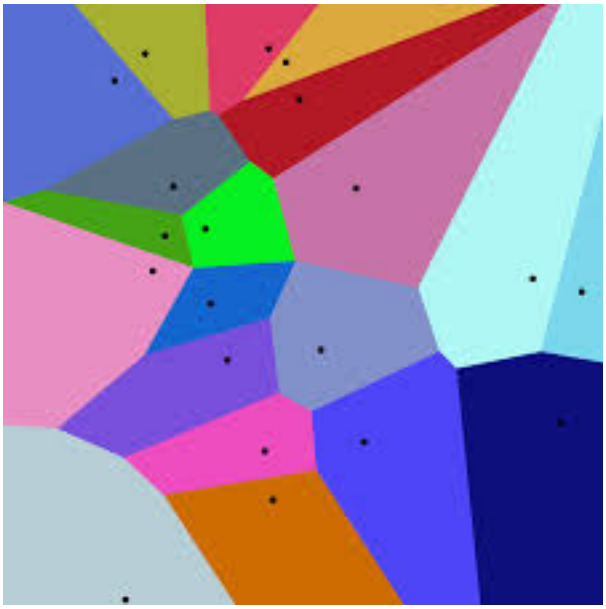


Figure 3.4: A voronoi mesh constructed between the particle points

is a defined border between the two particles, and thus for any two particles with differing densities, for example, there will be a discontinuous jump all along the face. These fixed types of connections with other cells created by mesh particles enable a definition of the mathematical operators on our predefined faces of the cells, increasing the accuracy of the approximations of the derivatives and presenting a valid advantage for this particular method.

The implemented solvers actually have another benefit, as they facilitate implicit mixing across the surface boundaries due to their averaging. If we recall how a Riemann solver functioned for our simple one-dimensional test case, we remember that the area around the discontinuity experienced an averaging of the two density values, which traveled outward into each cell. This in turn leads to an implicit production of entropy through the mixing of different fluxes in a single cell (as the characteristics from each cell it shares a face with will travel into them, see Springel's AREPO paper [25] for more details on this aspect of mesh-based codes). Mesh codes are therefore excellent at capturing discontinuities, ergo shocks, and additionally fluid mixing due to the implementation of Riemann solvers (this also explains the interest in creating a meshless method which also uses said solvers, in the hopes of achieving all SPH advantages and still having good shock and mixing capabilities).

To avoid the aforementioned stark deformations of the faces and the errors arising from the complex motions of fluid dynamics, one can implement a renormalization scheme. This could be achieved by for example requiring the individual cells of the mesh to have mass in a specified range, thereby preventing them from growing too large and distorted while also keeping the mass resolution reasonably constant, or by explicitly preventing sharp angles in their faces [25].

However, the implicit entropy production that facilitates the good mixing can lead to

overmixing as well. For fixed mesh codes such as ATHENA [26], the additional lack of Galilean invariance leads to a large downside in that it is unable to cope with bulk flows. This inability arises due to the Riemann solvers, as the conditions (and therefore the actual solution) of the Riemann problems are drastically altered. We assume for our normal problems that the reference frame of the boundary is fixed, so that we know how much material reached the border and can interact (recall our limitation of the timestep size in section 3.3, where we explicitly made sure this is given), but the application of supersonic bulk flow changes exactly this amount of material that can reach the border. Such supersonic bulk flows also happen to be necessary for the simulation of galaxy formations, a large downside for mesh-based codes. This can, however, be avoided for moving mesh codes such as AREPO so long as these are moving along with the flow [25], as then we have fit our reference frame onto the supersonic flow itself, such that the calculated amount of interacting material is correct again.

### 3.5.1 Formulation of fixed and moving meshes

Our derivation will closely follow the derivation of the equations of motion used for mesh-based codes as presented by Springel [25]. We begin with the Euler equations as a system of hyperbolic partial differential equations in a compact form we have

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F} = 0 \tag{3.24}$$

where $\mathbf{U}$ is a state vector for the main system variables density, velocity and total energy per volume (note that $u$ here is the specific internal energy and $e$ the specific total energy)

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho e \end{pmatrix} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho u + \frac{1}{2}\rho \mathbf{v}^2 \end{pmatrix} \tag{3.25}$$

and the tensor $\mathbf{F}$ is the flux function based on $\mathbf{U}$ and the equation of state 2.25

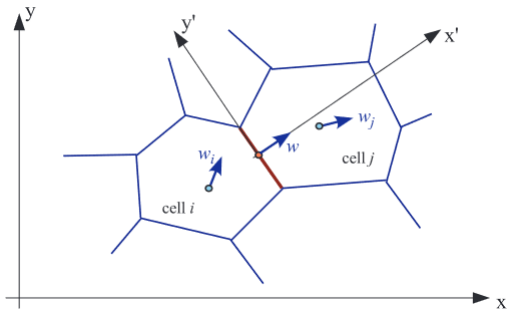



Figure 3.5: Sketch of the flux calculation along the face bordering cell i and j, whose position is entirely determined by the motion of the mesh-generating particles it is between. The flux may be estimated based upon a Riemann problem along said face

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho\,\mathbf{v} \\ \rho\,\mathbf{v}\mathbf{v}^T + p \\ (\rho e + p)\,\mathbf{v} \end{pmatrix} \tag{3.26}$$

The total conserved variables are then the volume integral of particle i over the state vector $\mathbf{U}$

$$Q_i = \begin{pmatrix} m_i \\ p_i \\ E_i \end{pmatrix} = \int \mathbf{U}\,\mathrm{d}V_i \tag{3.27}$$

However, we must consider that since we are creating a mesh upon which we would like to calculate our variables, we must transform to a coordinate system moving with the mesh-generating particles. Let $\mathbf{w}$ be the velocity at which each particle moves and we get the compact Euler equations in the form relevant for both mesh-based schemes and MFM (as we will see in section 3.6)

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\mathbf{F} - \mathbf{U} \cdot \mathbf{w}^T\right) = 0 \tag{3.28}$$

We now take these equations and integrate them over the volume of each cell $V_i$ (we have therefore split up our volume into smaller parts and integrated over each separately, hence this is finite volume method). Mesh-based codes then transform the volume integral over the flux divergence in our Euler equations into a surface integral between the discretized cells by using the Gauss' theorem. Let $\mathbf{n}$ be the normal vector on the effective particle surfaces $\partial V_i$ and we get

$$\frac{\partial \mathbf{Q}_i}{\partial t} = -\int_{\partial V_i} \left[\mathbf{F}(\mathbf{U}) - \mathbf{U} \cdot \mathbf{w}^T\right]\,\mathrm{d}\mathbf{n} \tag{3.29}$$

where we have also assumed the state vector to be sufficiently smooth so that we may invert the order of the integral and the partial time derivative for each particle separately. If $\mathbf{A}_{ij}$ is the area of the face between cells i and j and the velocity of each of the points along this boundary is $\mathbf{v}$ (compare figure 3.5 ), then the flux across this face from i to j is

$$\mathbf{F}_{ij} = \frac{1}{\mathbf{A}_{ij}} \int_{\mathbf{A}_{ij}} \left(\mathbf{F} - \mathbf{U} \cdot \mathbf{w}^T\right)\,\mathrm{d}\mathbf{A}_{ij} \tag{3.30}$$

This finally leads us to our Euler equations in a form immediately recognizable as being akin to the equations derived in Godunovs finite volume scheme 3.16

$$\frac{\mathrm{d}Q_i}{\mathrm{d}t} = -\sum_j \mathbf{A}_{ij} \cdot \mathbf{F}_{ij} \tag{3.31}$$

$$\overline{Q}_i^{n+1} = \overline{Q}_i^n - \Delta t_i \sum_j \mathbf{A}_{ij} \cdot \tilde{\mathbf{F}}_{ij}^{n+1/2} \tag{3.32}$$

With $\tilde{\mathbf{F}}_{ij}^{n+1/2}$ being the approximated fluxes from cell i to cell j averaged over a timestep. This equation is the fundamental form of the governing equations of motion for our simulations. As is remarked by Springel, a critical step is finding a good numerical estimate for these fluxes, as their accuracy directly determines the accuracy of our time evolution, and therefore a lot of numerical fluid dynamics literature has been dedicated towards this topic.

## 3.6 Meshless finite mass

As stated by Hopkins himself, the intent of the meshless finite mass scheme (and also of the meshless finite volume scheme) is to capture advantages of both the Eulerian mesh-based and the Lagrangian SPH methods. In MFM we find a method that is closest to the method presented in [14]. This means that we lack discrete borders, but rather construct a moving voronoi mesh with 'effective faces' between the particles on which the fluxes are calculated (like for moving mesh codes), but then calculate the state variables over smeared volume partitions partially overlapping with other particles (which is a property akin to those of SPH codes). As with a finite volume scheme and therefore also with the meshless finite volume method MFV (not discussed here as it closely follows the principles of MFM), the volume is partitioned into a finite number of cells. These, however, need not be completely disjoint as with mesh codes but rather will overlap and must be weighted. This means that on one hand we need Riemann solvers for the fluxes on the effective faces, but on the other hand we project those fluxes (that are localized on the faces) to the *particles* according to a distance-based weighting function. Referring to figure 3.6 we can see exactly how the methods vary.

The defining characteristic of MFM as opposed to MFV is the way the volume is partitioned. For MFV the total *volume* on either side of the border between two cells is kept constant, independent of the exact mass distribution between them, whilst MFM keeps the amount of *mass* constant. This means MFV partitions the volume, whilst MFM partitions the mass (as their names imply). For MFM this conservation of the mass on both sides leads to a volume that is then continuous by definition, as the initial particle masses are constant. Then, as here we keep the volume continuous, we must be discretizing the density instead, which means that we (just as with SPH) are then *smoothing* the density over one kernel length for MFM. This is not present in MFV (as for MFV the mass not the volume is continuous), which then means we do not observe the smoothing here. One can observe this with the fluid mixing tests, for example, such as the KHI test, where the small scale dynamics are smoothed out for MFM once we reach length scales comparable with the kernel length.

Finally, as all conservative methods implementing a Riemann solver have errors when the kinetic energy dominates over the thermal energy which is the case for cold supersonic cosmological structures with very high Mach numbers (these arise as then the numerical errors in the thermal energy start becoming large relative to the total thermal energy), MFM incorporates the 'dual energy' formalism and energy-entropy switches where the

specifics can be taken from [8].

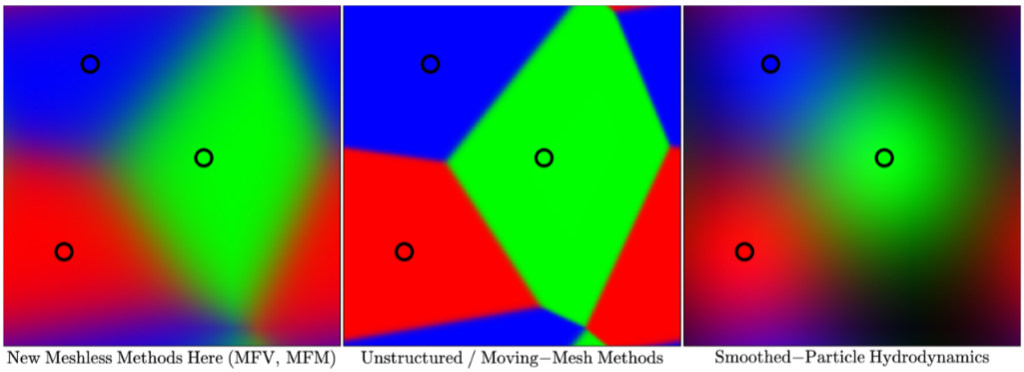



Figure 3.6: The different approaches characterized by how they calculate the motion of the mass particles, i.e. by how they partition the volume. *Left* are the MFM and MFV methods: the domain associated with each particle for flux calculation is not spherical, despite it being so for the weighting function defined by the kernel for state variable calculation through the equations of motion. The border is smoothed over a typical kernel width. The *center* shows traditional mesh partitions with sharp boundaries, presenting the limit of MFM with an infinitely sharply peaked kernel function. The equations of motion are solved along the surface of the faces as opposed to at the location of each particle. On the *right* we find the SPH method, where the state variables of the neighboring points are used in the calculation of the equations of motion of each point according to a weighted average based upon the kernel function.

### 3.6.1 Formulation of MFM

Here then the meshless equations of motion as utilized by for example Gaburov & Nitadori, Hopkins and Lanson & Vila ([6],[8] and [14]) are implemented. We provide only a rough derivation outline, for the specifics see the above papers, or for a full derivation we suggest specifically refering to Lanson & Vila [14].

Just as with the moving-mesh derivation, the equations we start from are the compact form of the Euler equations, or rather with their form in a frame moving with velocity $\mathbf{w}$ as in equation 3.28. We then multiply these with a test function $\phi$, integrating over the space and using a partial integration (assuming the fluxes or the test function vanish at infinity) so that we arrive at a new integral. To solve this arising set of differential equations we discretize the integral volume of space (as stated above) into a finite number of cells with associated specific characteristic lengths $h$ as with SPH, finally arriving at

$$0 = \sum_i \left( \phi_i \frac{\mathrm{d}}{\mathrm{dt}} (V_i \mathbf{U}_i) - V_i \mathbf{F}_i \cdot (\Delta \phi) \right) \tag{3.33}$$

where the inner product is to be taken at the spatial position $x_i$ of particle i. We determine that an approximation for the gradient of the test function is needed, which should be

accurate to the second order so that we may preserve our efforts in keeping all preceding approximations accurate to this same level. Utilizing the gradient estimator as presented in gizmo [8] and additionally defining $\tilde{F}^{\alpha}_{ij}$ as the approximate solution for the fluxes between cells i and j given by the solution of the Riemann problem (which therefore incorporates both sides of the flux and satisfies $\tilde{F}^{\alpha}_{ij} = -\tilde{F}^{\alpha}_{ji}$ by definition) we arrive finally at

$$\frac{\mathrm{d}}{\mathrm{dt}}(V_i \mathbf{U}_i) + \sum_j \tilde{F}^{\alpha}_{ij}(V_i \tilde{\psi}^{\alpha}_i(\mathrm{r}) - V_i \tilde{\psi}^{\alpha}_i(\mathrm{r})) = 0 \tag{3.34}$$

where $\tilde{\psi}^{\alpha}_i(\mathrm{r})$ is a function based on the fraction $\psi_i(\mathrm{r})$ of the total volume associated with the particle i according to a distance based weighting function (quite like the methods used in SPH). As stated earlier, this follows the form of our Godunov-type finite-volume equation 3.16 (if we define $A_i \equiv V_i \tilde{\psi}^{\alpha}_i(\mathrm{r}) - V_i \tilde{\psi}^{\alpha}_i(\mathrm{r})$), but fundamentally differs in that the volume integral 3.27 for the conserved variables used in the derivation is not transformed into a surface integral (as it would be in for example AREPOs equations of motion, see 3.30) but rather is partitioned up via a weighting function (as is the case for SPH). It however also differs from SPH in that whilst the conserved state variables aren't, the *fluxes* between particles are calculated on the particle faces, i.e. over a surface integral as with mesh codes.

To summarize, MFM represents a meshless adaptable code that has characteristics for the calculation of the time evolution that are akin to both SPH and mesh-based codes: the fluxes are calculated on the faces, and the state variables in the volumes.

### 3.6.2 Implemented kernels

As the choice of kernel has a large impact on the results, we present the implemented kernel here. For all our tests, either the traditional cubic spline by Monaghan and Lattanzio [17] or the quintic spline as implemented by Hopkins [8] are used, where the cubic is run with 32 neighbors whilst the cubic uses a higher number of up to 128. The cubic spline in three dimensions is given by (for $r = |\mathbf{r} - \mathbf{r}'|$ and $q = r/h$ the distance to the neighbor in terms of h)

$$W(q,h) = \frac{8}{\pi h^3} \begin{cases} 1 - 6q^2 + 6q^3 & 0 \leq q < \frac{1}{2} \\ 2(1-q)^3 & \frac{1}{2} \leq q < 1 \\ 0 & q \geq 1 \end{cases} \tag{3.35}$$

where the coefficient changes for the cubic spline in two dimensions.

The quintic spline is given by Hopkins quintic spline as

$$W(q,h) = \frac{3^7}{40\pi h^3} \begin{cases} (1-q)^5 - 6(\frac{2}{3}-q)^5 + 15(\frac{1}{3}-q)^5 & 0 \leq q < \frac{1}{3} \\ (1-q)^5 - 6(\frac{2}{3}-q)^5 & \frac{1}{3} \leq q < \frac{2}{3} \\ (1-q)^5 & \frac{2}{3} \leq q < 1 \\ 0 & q \geq 1 \end{cases} \tag{3.36}$$

so for the cubic and quintic spline we note that no points further than one kernel length h away are counted. Note that in practice $h$ will depend on the particle i. We will additionally use the Wendland $C^4$ kernel for select cases, for which the theoretical background was first devised by Wendland in [31].

# Chapter 4

# Tests

This chapter provides a brief summary of the physics of the test cases. The governing equations for most of the problems are given in chapter 2, with some specifics elaborated upon here where necessary. Additionally, we describe the initial conditions of the problems, that are those provided by Hopkins with his gizmo distribution, and which kernels were used for which test, before then in chapter 5 presenting our simulation results. All initial conditions values are given in the Gadget internal units, and all tests will be plotted with eqaul mass particles (one could, as discussed in chapter 3, instead use more particles for denser areas).

## 4.1 Tests in equilibrium: the Gresho vortex

A simple test to begin with is the Gresho vortex, which is a vortex of constant density and an initial azimuthal velocity distribution in form of a triangle and was first conceptualized by Gresho and Chan in 1990 [9]. The advantage of this test is that the analytical answer is known, namely as the test is initialized in equilibrium it should stay in equilibrium. This means that the conservation properties of the code can be tested, so their preservation of symmetry and angular momentum - especially the latter - as the vortex is constantly rotating. By looking at the preservation of symmetry on the other hand one can specifically test the properties of the different mesh-construction methods (i.e.: voronoi mesh versus uniform divisions in specific spatial regions for example, or uniformly-spaced static grids) for mesh-based codes. These codes are also known to do very well with this test, while SPH is known to struggle greatly here. This is due to the fact that SPH generates large amounts of noise from the E0 error and also suffer from what Hopkins [8] describes as 'volume partition noise', which can quickly degrade the vortex.

The vortex itself is an area of constant density, where the velocity varies depending on the radius. This is constructed in such a way that the vortex is in hydrodynamical equilibrium, namely in pressure equilibrium with the surrounding medium. The resulting azimuthal velocity profile rises linearly until a specified radius, and then decreases linearly until finally reaching the static outer areas.

Our test is initialized in a box of sizes 1x1 in two dimensions. Throughout the box we will find a constant density $\rho = 1$. It should, however, be mentioned that the regions of the vortex itself (for $0 \leq r \leq 0.4$), with r defined as the distance from the point to the center of the box (x=0.5, y=0.5), are simulated by equidistant points of nearly perfect constant density $\rho = 1$, whilst the outer regions $r > 0.4$ have their constant density initialized through a random distribution around $\rho = 1$ with fluctuations of up to $\pm 0.2$. We then calculate the azimuthal angle (as polar coordinates are the natural choice for a vortex) to be

$$\phi(x,y) = \begin{cases} atanh\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \frac{1}{2}\pi\frac{y}{|y|} & \text{if } x == 0 \\ atanh\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \,\text{and}\, y \geq 0 \\ atanh\left(\frac{y}{x}\right) - \pi & \text{else} \end{cases} \tag{4.1}$$

where x and y are the particle positions. The azimuthal velocity is then the velocity component along the above angle We then have an azimuthal velocity profile following Hopkins implementation as such

$$v_\phi(r) = \begin{cases} 5r & \text{if } 0 \leq r < 0.2 \\ 2 - 5r & \text{if } 0.2 \leq r < 0.4 \\ 0 & \text{else} \end{cases} \tag{4.2}$$

This describes a rotating vortex with a radial velocity of zero everywhere. As the state is in equilibirum, the analytical solution of the distribution of the azimuthal velocity is the form described above, namely a linearly rising line until $r = 0.2$, followed by an equally steep linear decline until $r = 0.4$ until finally staying constantly zero after (with regards to the radius). It should also be noted that this is one of the two tests for which we will be using an adiabatic coefficient$\gamma = 1.4$. This has historical reasons and is the adiabatic coefficient for diatomic gas, which can be the coefficient used for interstellar gas clouds (see Spaan and Silk's analysis on such clouds for more info on when this coefficient is accurate [22]). Additionally, we can imagine that this test can be seen as a rough approximation for rotating structures in equilibrium, so might be used more sophisticated versions to simulate for example rotating galaxies.

## 4.2 Shocks: Sod shock tube and Sedov-Taylor explosion

Shocks are a prominent way to test a code's reliability with handling contact discontinuities and large Mach numbers, and provide a good measure of their accuracy when dealing with large differences in density overall. Mesh-based codes tend to have an inherent advantage with these tests, as they consist in essence of Riemann problems and the Riemann solvers of the mesh codes are therefore specifically designed to capture such flows. This leads to

less diffusion than is experienced with SPH, although fixed-mesh solutions are known to substantially suppress the jumps as well. These tests are important, as contact discontinuities in density and especially large Mach numbers are commonplace for astrophysics, if we think of for example clouds of different materials coming in contact with each other, or a supernova explosion which moves outwards rapidly.

### 4.2.1 Sod Shock Tube

The Sod shock tube is one of the most common tests for code accuracy with shocks and consists of two regions of differing densities and pressures initially at rest. This is therefore an example for a one-dimensional Riemann problem. Mathematically then, the initial state is described by

$$\rho(x, t = 0) = \begin{cases} \rho_1 & \text{if } x < x_0 \\ \rho_5 & \text{if } x \geq x_0 \end{cases} \tag{4.3}$$

$$p(x, t = 0) = \begin{cases} p_1 & \text{if } x < x_0 \\ p_5 & \text{if } x \geq x_0 \end{cases} \tag{4.4}$$

where for our case the initial densities are $\rho_1 = 1$ and $\rho_5 = 0.25$. The initial pressures are

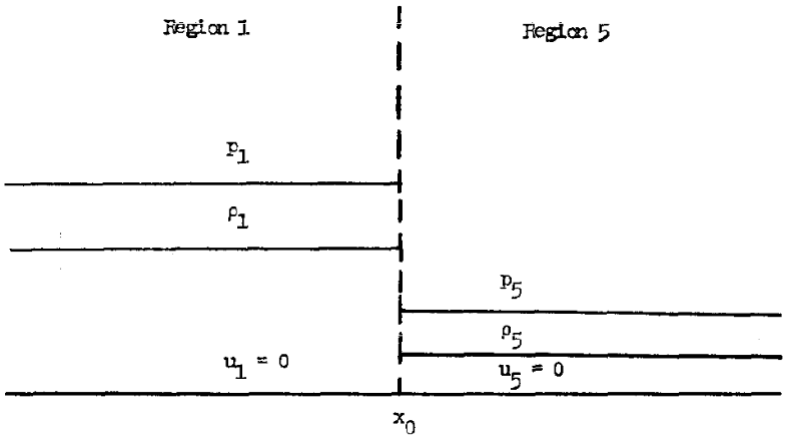



Figure 4.1: The initial conditions of the 1D Sod shock problem for t=0. We see that the velocities (described as u here, not to be confused with the internal energy) are equal to zero on both sides, so the particles are initially all at rest, while pressure and density are larger on the left than on the right

$p_1 = 4$ and $p_5 = 0.7$. With $\gamma = 1.4$, this is our second test with the adiabatic coefficient of diatomic gas.

As the test evolves the areas begin to mix, forming three distinct regions inbetween the two starting regions on the left and right, each with varying densities, pressures and velocities. The exact form can be taken from figure 4.2. Between the regions 3 and 4 we

see the emergence of a contact discontinuity in the density profile, and between regions 4 and 5 a forward (to the right) moving shock wave.

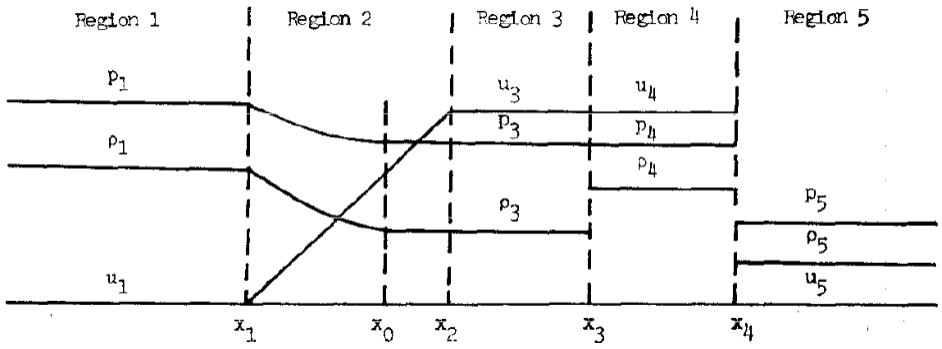



Figure 4.2: The analytic evolution of the Sod shock tube for a time t>0. We can see that five distinct regions form around the initial contact discontinuity at $x_0$, where the left- and rightmost regions describe our two initial regions. It should be noted that while this is the solution presented by Gary Sod in his original paper, the density in region 4 does not actually increase as shown here but rather decreases to a value between those of region 3 and 5.

Let us now look at the derivation of only the most basic relations following the one presented by Dullemond [5], as a full derivation is quite tedious and as such will not be performed here. We start by looking at the (numerically derived) qualitative form of the five regions. Regions 1 and 5 describe the initial states, 3 and 4 have constant pressure, density and velocity, whilst for region 2 all values vary depending on x.

If we then turn our attention first to the contact discontinuity between 3 and 4, we note that as both regions move with the same velocity $v_3 = v_4 \equiv v_c$ we can conclude the speed of the contact discontinuity to be identical and its position at a given time to be at $x_3 = v_c t$.

Now considering the regions 4 and 5, we see that as the shock wave propagates to the right, the velocity behind it from view of the laboratory is $v_4$ , and ahead of it is $v_5 = 0$. Let us consider then a coordinate system moving along with the shock, ergo with the shock at rest. The speed of the gas particles infront of the shock in region 5 is then $v_5 - v_s$ with $v_s$ being the speed of the shock in the laboratory. The speed behind the shock is equivalently $v_4 - v_s$. Mass conservation for the shock system 2.36 then dictates

$$(v_5 - v_s)\rho_5 = (v_4 - v_s)\rho_4 \tag{4.5}$$

$$v_s\rho_5 = (v_s - v_4)\rho_4 \tag{4.6}$$

$$v_s = v_4\frac{\rho_4}{\rho_4 - \rho_5} = v_4\left(1 - \frac{\rho_4}{\rho_5}\right) \tag{4.7}$$

From this and the density ratios of pre- and post-shock 2.63, we can derive

$$\begin{aligned} v_4 &= (p_4 - p_5)\sqrt{\frac{2}{\rho_5(\gamma+1)(p_4 + \frac{\gamma-1}{\gamma+1}p_5)}} \\ &= v_c \end{aligned} \tag{4.8}$$

Turning our attention now to the region 2, which describes an expansion wave, we know that the left border of the region is propagating toward the left with the local speed of sound 2.34 given by the pressure $p_1$ and density $\rho_1$ immediately at the border

$$x_1 = -\sqrt{\frac{\gamma\, p_1}{\rho_1}} \tag{4.9}$$

The gas velocity can be derived (see for example Hawley's derivation [10], however note that he uses different variable definitions with inter alia our $m$ being his $\mu$) as

$$v_{2border} = \sqrt{\frac{(1-m^4)p_1^{\frac{1}{\gamma}}}{m^4\rho_1}}\left(p_1^{\frac{\gamma-1}{2\gamma}} - p_2^{\frac{\gamma-1}{2\gamma}}\right) \tag{4.10}$$

where $m^2 \equiv \frac{1+\gamma}{1-\gamma}$.

We can then solve the location of the border between region 2 and 3 via equalizing the velocities, so by setting $v_{2border} = v_3$ ( $= v_4$ for which we derived a value in equation 4.8) and then solving this numerically to get a value for $p_3 = p_4$. We then know the gas velocities $v_1, v_3, v_4$ and $v_5$ (as the velocity $v_{2border}$ is only valid on the border to region 3, we don't actually know it fully yet), the pressures $p_1, p_3, p_4, p_5$ and the densities in our starting regions 1 and 5. We need all values in region 2 then, plus the densities for region 3 and 4. We first deduce the density in region 4 from 4.8, and then the density in region 3 by realizing that as the shock front is moving to the right, gas to the left of the contact discontinuity has never gone through a shock front (only the gas that was in region 5 has). Therefore the entropy must be equal to that of region 1, and using the polytropic gas law 2.2 $pV^a = const$ for an isentropic process, ergo where $a = \gamma$ the adiabatic coefficient, we can conclude $p\rho^{-\gamma} = const$, or

$$\rho_3 = \rho_1\left(\frac{p_3}{p_1}\right)^{\frac{1}{\gamma}} \tag{4.11}$$

With regions 1,3,4 and 5 now fully known, we just need the velocity, density and pressure for region 2, which we give without derivation:

$$v_2(x,t) = (1-m^2)\left(\frac{x}{t} + c_{sound,1}\right) \tag{4.12}$$

$$\rho_2(x,t) = \left(\frac{\rho_1^\gamma}{\gamma p_1}\left(v_2(x,t) - \frac{x}{t}\right)^2\right)^{\frac{1}{\gamma-1}} \tag{4.13}$$

$$p_2(x,t) = p_1\left(\frac{\rho_2(x,t)}{\rho_1}\right)^\gamma \tag{4.14}$$

All relevant values for our system, for example the sound speeds or internal energies, can be derived from the density, pressure and gas velocity of the five regions, which in turn one can deduce via the above equations entirely from the values for the density and pressure of our initial two states.

### 4.2.2 Sedov-Taylor explosion

The Sedov-Taylor explosion, or Sedov blast, is another one of the test cases where the analytic solution is fully known. The simulation starts with a constant density throughout the box $\rho_1$, with a small point in the center being given high starting energy at the initial time. For this specific test we use a Wendland $C^4$ kernel, with a correction presented by Dehnen and Aly [4] instead of a cubic or quintic kernel, in addition to a wake-up function as opposed to a slope limiter. This in turn requires a greater number of neighbors (around 200) but produced slightly better results for us than the quinitc or cubic kernel did.

As the system evolves with time, the high energy in the center creates a spherically symmetric shock wave of higher pressure, temperature and density that travels radially outward. The difference to a normal wave is the very sharp change in gas properties, nearly comparable with a discontinuous jump. This disturbance moves outward faster than the local speed of sound within the gas, thereby preventing the matter from parting before it and quickly building up a region of high pressure (as described by Anderson [1]).

Because, as we have stated, this area is comparable with a discontinuous jump, we have a situation as presented in the section 2.3. Our governing equations are then those we derived there

$$\text{continuity :} \qquad \rho_1 v_{x1} = \rho_2 v_{x2} \tag{4.15}$$

$$\text{momentum :} \qquad p_1 + \rho_1 v_1^2 = p_2 + \rho_2 v_2^2 \tag{4.16}$$

$$\text{energy :} \qquad \tfrac{1}{2} v_1 + e_1 + \frac{p_1}{\rho_1} = \tfrac{1}{2} v_2 + e_2 + \frac{p_2}{\rho} \tag{4.17}$$

resulting in the evolution of the two phases as in 2.63 and 2.66

$$\frac{\rho_2}{\rho_1} = \frac{\gamma + 1}{\gamma - 1} \tag{4.18}$$

$$\frac{p_2}{p_1} = \frac{2\gamma M_1^2 - (\gamma - 1)}{\gamma + 1} \tag{4.19}$$

For this test case we have $\gamma = 5/3$, meaning that in an ideal scenario we would see the immediate post-shock density being exactly four times as high as the pre-shock density. If we then plot the density over the radius we should expect to see a constant density for large radii (equal to our initial constant density), which then (ideally) discontinuously jumps up to our aftershock maximum value and then starts decreasing again with smaller radii, as the particles are relaxing after the shock and spreading out (unlike the shock front where they are bundled together). Additionally, the density should be the same for equal radii,

as the blast propagates radially outward. We will see this when plotting the density over slice of the z axis, so over a select x-y plane.

Our initial conditions describe a box of size 6x6x6 with a constant density of $\rho = 1.23 * 10^7$ throughout. The central point starts with energy of around 0.00503 in Gadget's internal units, which corresponds to the average energy in a supernova $1 * 10^{51}$ erg. We can therefore use this test to simulate the remnants of supernovas and see their distribution, depending on which we can stipulate their initial energy. But it also helps assess how a code would deal with large-scale interactions, where high Mach numbers are commonplace.

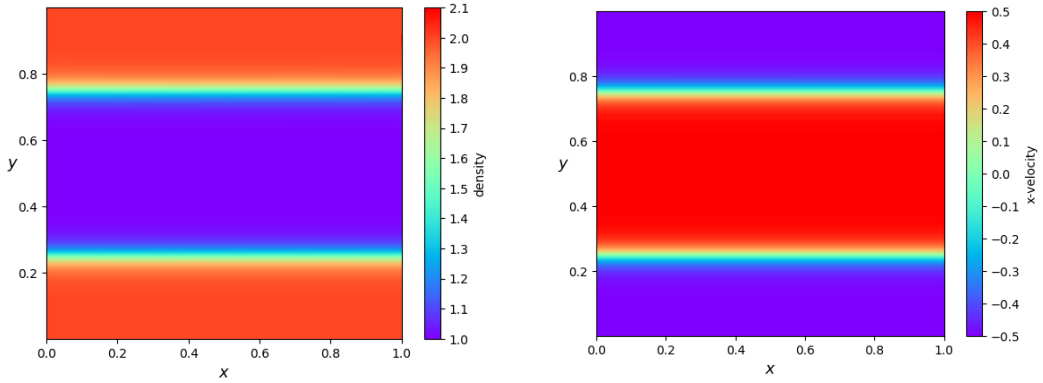
## 4.3 Fluid mixing: KHI and the "Blob" test

Another set of commonly used tests encompasses the code's ability to accurately represent the mixing process of fluids (this is particularly relevant in larger scale structure formation). The issues SPH may have with such a situation are evident, as the mixing at fluid boundaries is difficult to resolve due to the leading error in the momentum equation (E0 error) and the local mixing instability. The former suppresses low amplitude modes, and the latter occurs due to attempted mixing on the kernel scale being prevented by entropy conservation. This arises as mixing would be an increase in entropy, so when each particle tries conserving entropy they 'fight' against mixing with other particles, and this leads to an increase in pressure and a high surface tension (see Agertz [2] excellent in-depth explanation for more). The result is, that the expected increase in entropy through the mixing, which is supposed to lead to energy being freed (which facilitates said mixing in the first place), is only present for mesh-based codes. For SPH one must artificially increase the mixing capabilities by adding artificial viscosity/conductivity terms, but the former E0 error persists in all variants of SPH and can only be lessened by increasing the number of neighbors (for an overview over most plausible solutions for resolving the mixing errors of SPH see Read [21]).

However, there are also issues for mesh-based codes when simulating fluid mixing, as assigning the entire fluid an additional velocity comparable to the shear velocities between the individual parts of the fluid leads to large errors. Put simply, this means that the mesh-based codes struggle with bulk velocities. This again is unfortunate, as bulk velocities are omnipresent in astrophysical situations, especially those involving structure formation. It should additionally be obvious, that the codes capabilities to mix gas of differing densities and velocities should be relevant in a lot of situations for astrophysics.

### 4.3.1 Kelvin-Helmholtz instability

The Kelvin-Helmholtz instability occurs when there is a velocity difference across the boundary of two fluids. This causes them to rub against each other, leading to a mixing between them and then transitions into turbulent flow. This specific scenario will be initialized without gravity (there is also the variation with the denser fluid at the bottom and gravity acting on the system to facilitate mixing) and with a central area of a lower density fluid




(a) The distribution of the density with the smoothing layers.

(b) Distribution x-component of the velocity.

Figure 4.3: The initial conditions of our KHI test

moving to the left, where the top and bottom high density areas are moving to the right, thereby resulting in a velocity difference at two boundaries. Then a sinusodial disturbance is introduced along the boundary (which has non-zero thickness), resulting in the instability and finally turbulent flow.

The governing physics - as long as the system is stable - are dictated by the Navier-Stokes equation, or for inviscid flow by the Euler equations. If gravity were present, one could additionally simplify the equations by linearizing them and arriving at the Taylor-Goldstein equation. As shown by Kundu in [12], a setup with differing velocities across an interface always has wave modes (for large enough wave numbers k) that lead to instability. For our specific case, as described by Hopkins, we initially have an inviscid system. This however leads to the non-linear structe (i.e. in the swirls) lacking a converged solution, so to avoid this and obtain a define-ably converged solution we must ensure that we add finite viscosity to the system. As it is however, there is no analytical solution to our problem once the system transitions to turbulent flow. Characteristic swirls are formed in the non-linear growth phase, where the stability finally collapses, through the continuously increasing overlap of the phases into each other before they are rolled up (due to the motion of the surrounding fluid). Later into the non-linear phase the rolls grow large enough to finally overlap with each other forming larger, more complex structures.

The test case is initialized in a periodic two-dimensional box of size 1x1. As described above and as can be taken from figure 4.3, we then have a central area with lower density $\rho_{low} = 1$, moving with velocity $v_x = 0.5$ to the right in direction of x (in Gadget's internal units). The central area is located from around $y = 0.25$ to $y = 0.75$. We see the smoothing layer at these values, which take on an exponential interpolation between the inner and outer layers leading to a smoothed transition. The outer areas have double the density, so $\rho_{low} = 1$, and move exactly opposite in x-direction with velocity $v_x = -0.5$, leading to a total shear velocity of 1. As mentioned we then seed a small (of up to 0.001) disturbance velocity in y-direction, whose amplitude follows a sinusodial pattern in x-direction.

The internal energy is distributed 'inversely' to the density (and therefore balances out the pressure between the layers), as the central area has a higher energy of $e_{high} = 4$ whilst the outer areas have $e_{low} = 2$, with a non-zero thickness between the layers which gradually transitions between their energies.

### 4.3.2 The blob test

To see the impact of the fluid mixing instabilities (KHI and the Rayleigh-Taylor instability which is not discussed here) in a more 'realistic' situation we turn to the blob test. Here we observe the evolution of a central cloud, or 'blob', of uniform high density which is placed into a moving medium of constant, lower density. The phases are in pressure equilibrium. As the outer fluid phase moves past the blob, the fluid mixing instabilities tear away at the surface of the blob, much like what happens when a galaxy passing another strips it through tidal interactions. A bow shock is formed (as the blob forms a sufficiently blunt surface to avoid oblique shocks) along whose boundary the instabilities form, with additional shocks resulting from the periodicity (i.e. from multiple passes of the outer medium past the blob). Such bow shocks can for example be observed in astrophysics when a magnetized interstellar gas cloud passes a planet or star with a magnetic field, upon which along the boundary the bulk velocity of the gas drops from 'supersonic' to 'subsonic'.

We initialize a box size of 2000x2000x6000 with a constant density throughout, where the blob density is ten times higher than the ambient density $\rho_{blob} = 3 * 10^{-7} = 10 * \rho_{ambient}$. The blob is placed slightly to the left of the center on the large z-axis, so as to ensure that the tail, which will form through stripping of the blob, stays mostly in the box without coming back in over the periodic border. This places the blobs center therefore at (x=1000, y=1000, z=2000). The exact set up can be taken from figure 4.4, where we can additionally see that the radius of the blob is 197. These initial conditions are then roughly equal to those of the Wengen test suite as described by Agert et al. [2].

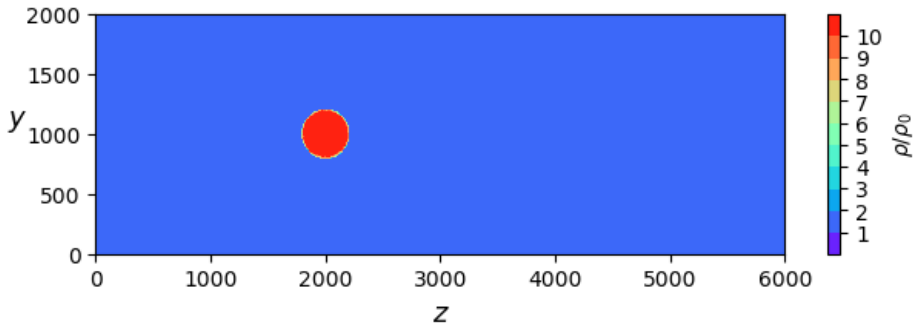



Figure 4.4: The initial conditions of the blob test. We see the density distribution in a slice for $0 \leq y \leq 2000, 0 \leq z \leq 6000$ and $x = 0$.

To ensure pressure equilibrium the temperature and therefore the internal energy of the blob is ten times lower than that of the ambient medium, which we can see in the initial

conditions in figure 4.4 and can confirm via the equation of state 2.25

$$p_{blob} \stackrel{!}{=} p_{ambient} \tag{4.20}$$

$$e_{blob}(\gamma - 1)\rho_{blob} = e_{ambient}(\gamma - 1)\rho_{ambient} \tag{4.21}$$

$$\frac{e_{blob}}{e_{ambient}} = \frac{\rho_{ambient}}{\rho_{blob}} = \frac{1}{10} \tag{4.22}$$

As this test requires good fluid mixing capabilities and tests for not only KHI but also Rayleigh-Taylor instabilities, we can expect in the results to see SPH struggle a bit. The resulting surface tension due to entropy conservation prevents the mixing from happening on the scales of the other codes: We expect a result which mostly flattens the blob, leaving larger parts intact than with MFM (although as we are using the more modern PSPH the effects should not be as severe as they would be for traditional SPH). For MFM we expect equally good mixing capabilities as with the KHI test itself, so that the cloud is completely mixed, i.e. destroyed, through the disruptions at the surface within a few cloud-crossing timescales.

# Chapter 5

# Results

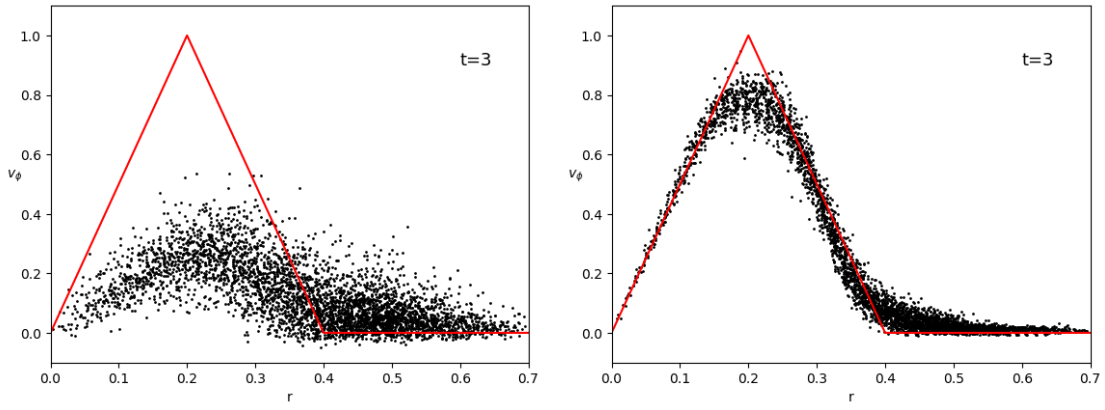
Here we present the results of our simulations, all run with both MFM and the SPH implementation as utilized by Hopkins [8]. We find good agreement with the results in the original MFM paper, namely overall good perfomance with fluid mixing in addition to good shock resolution. It should be stated in advance, however, that we do not manage to reach the same level of optimisation as the original paper.

## 5.1 Gresho vortex

As stated before in section 4.1, the analytical solution to the Gresho vortex has the vortex staying constant in time, perfectly equilibriating the inward and outward pressure. However, both methods do not perfectly conserve the system due to numerical inaccuracies and other instabilities/inaccuracies.

As expected, the SPH code struggles significantly with preserving the form of the vortex as it generates a large amount of noise. This can be seen clearly in figure 5.1: after around one orbit of the vortex, the original velocity distribution has been degraded significantly and the peak moved slightly outward, only reaching an average peak velocity of around 40% of the original value. The noise stems from both the E0 error being significant here, as well as the 'volume partition noise' described by Hopkins. The latter arises from the constant recalculation of the effective particle volumes in this test (as they are constantly experiencing a shear motion), leading to a noise for all fields depending on the volume (such as pressure). Furthermore, we see the artificial viscosity switches causing the particles to diffuse strongly outward, leading to a large amount of particles gathering at radii larger than 0.4. Additionally, we see a strong deviation from the original state already in the central area, with nearly no particles having their inital azimuthal velocity here.

In contrast, the MFM code manages to conserve an average peak velocity of around 80% of the original value. Especially well conserved are the areas toward the center of the vortex, where we can see barely any particles falling significantly out of the line described by the analytical solution. We do, however, observe as Hopkins does the same significant noise around the peak and especially for the transition into the static outer areas, which




(a) SPH gresho vortex (b) MFM gresho vortex

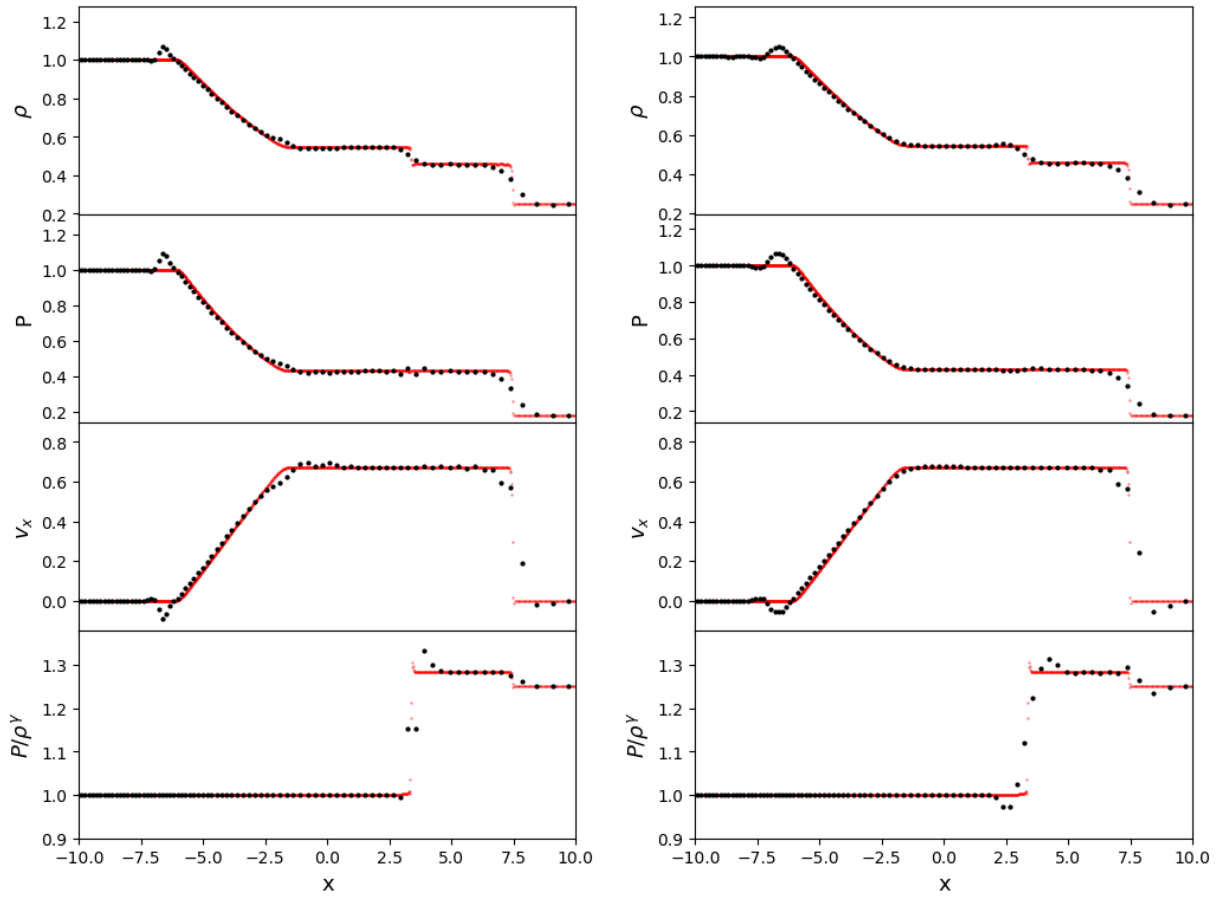
Figure 5.1: Comparison of the preservation capabilities of MFM and SPH via the Gresho vortex. We plot the azimuthal velocities versus the radius at t=2.5, with a resolution of all the $64^2$ simulated particles. The analytical solution is seen plotted with the red line.

he attributed to both a milder form of the 'volume partition noise' and the usual grid noise stemming from the motion/deformation of the effective faces used to calculate the fluxes.

## 5.2 Sod shock

The results of our shock tube test can be taken from figure 5.2. We observe that both codes behave similarly close around the converged solution. We see slight bulges before the transition at the front from constant density, pressure and velocity into the linear phases (x=-6). The bumps are present for MFM even with usage of equal mass particles. They start a little sooner for MFM than SPH but exhibit a smaller amplitude. These bulges are also seen around the contact discontinuities in the entropy at x=3 and x=7.5.

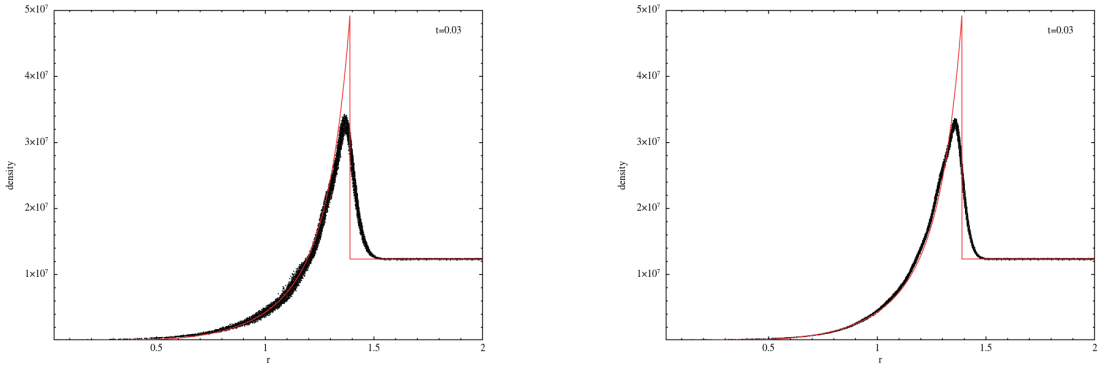
We also see a similar 'blip' in the pressure that is noted by Hopkins around where the first contact discontinuity is located (x=3), although it seems to be less defined for us for MFM and is instead best noticeable in SPH. The discontinuous jumps around x=3 and x=7.5 are smoothed visibly for both implemented codes, so as expected we do not capture the shocks as well as mesh-based codes can. Interesting to note here is the jump in the entropy at x=7.5, as for SPH we see just a smooth interpolation whilst for MFM the peaks before and after are still present as with the jump at x=3. Looking at this first discontinuity on the other hand, we see that MFM has bulges before and after of around equal amplitude, whilst for SPH the bulge before the jump is barely present, in contrast to the bulge after which is much more prominent than the one in MFM. Another interesting area to note is the transition from the linear to the constant phase around x=-2: We see SPH fluctuate visibly around the converged solution, while MFM on the other hand captures this area really well.




(a) SPH Sod shock tube

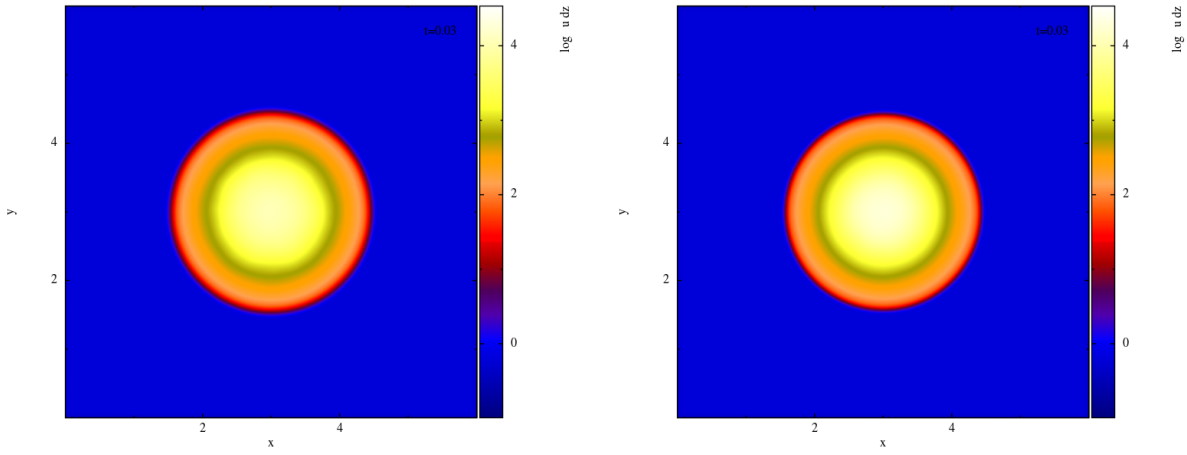
(b) MFM Sod shock tube

Figure 5.2: Comparison of the shock resolving capabilities of MFM and SPH. Plotted here are the density, the pressure, the velocity in x-direction and finally the entropy measure as presented in section 2.1.3. The black dots represent singular particles, whilst the converged solution of a high-resolution test is plotted in red. Both tests are plotted for the time t=5 at a resolution of 320 particles. We note that as we are working with equal mass particles here the particle density and therefore by nature the resolution of the solutions is dependent on the density itself, i.e. higher density leading to higher resolution for *both* MFM and SPH. This is therefore an excellent example of the Lagrangian nature of SPH and MFM.




(a) SPH Sedov blast (b) MFM Sedov blast

Figure 5.3: A comparison between the MFM and SPH sedov shocks. Whilst both reach similar peak hights, our SPH results on the left show a much higher diffusion than the equivalent MFM solution on the right. Plotted here is the density over the radius at time t=0.03 at a resolution of around 2.1 million particles.




(a) SPH Sedov blast (b) MFM Sedov blast

Figure 5.4: Plotted is the distribution of the logarithim of the internal energy over a slice at z=0 for t=0.03. We see good agreement between SPH and MFM, and both show excellent radial symmetry.

## 5.3 Sedov-Taylor explosion

We then turn to our second shock test, the Sedov-Taylor explosion. As can be taken from the radial density profile in figure 5.3, we find slightly different behavior in both our MFM and SPH implementations compared to the original MFM paper. It should be mentioned that this is the only test for which we employed a pressure-energy scheme for SPH, as opposed to the usual pressure-entropy scheme. This is due to the fact that for the usual scheme we found the shock to be moving ahead slightly faster toward the outside, i.e. with a higher initial energy. We speculate that this is due to the smoothing of pressure creating some additional $p\mathrm{d}V$ terms in the entropy, which would then increase the starting internal energy (as for a pressure-entropy scheme we calculate the other state variables from the pressure and the entropy). These effects were not present for the pressure-energy scheme, hence our choice here.

If we then turn back to the figure at hand, we find the jump coefficient, i.e. the factor between pre- and post-shock density, to be $\rho_2/\rho_1 \approx 2.72$ for MFM, whilst we calculate the one for SPH to be $\rho_2/\rho_1 \approx 2.77$. We observe additionally that we can further increase the jump factor for SPH by lowering the artificial viscosity. By doing this we could reach factors of around 3.3, which is then much closer to the analytical solution of 4 at the expense of a large increase in noise.

This means that while we observe the same suppression of the peaks for MFM and SPH, we actually found the suppression for MFM to be more severe than the one experienced by SPH, where even when we significantly increase the viscosity to reduce the noise (as is the case for figure 5.3) we still found a larger jump factor than with MFM. This is behavior opposite to that found by Hopkins, as there the MFM peak suppression was significantly smaller and therefore found MFM yielding jump factors much larger than those even for our most noisy SPH results (up to 3.5).

The noticeable peak dampening notwithstanding, we find MFM to produce a very good curve which closely follows the analytical solution with minimal noise, if any. Considering the figure shows a resolution of $128^3$, the lack of noise is quite impressive. On the other hand, even if we utilize high viscosity to reduce noise, we still find SPH to be noticeably more noisy than MFM.

If we then look at the logarithmic inner energy distribution in the x-y plane in figure 5.4, we find both the MFM and SPH codes to produce very compact and symmetrical rings. Both exhibit minimal noise here, with the only discernible difference being that the energy in the center dissipates very slightly more for SPH than for MFM. Aside from this both agree remarkably well with each other, and show excellent overall radial symmetry.

## 5.4 Kelvin-Helmholtz instability

The KHI test famously presents a large challenge for SPH codes, albeit not as severly as for example the Gresho vortex does. Mesh-based codes are known to work well here, so we expect similar results from MFM (as for this test the fluxes are most relevant). We will

compare here first the results of when the rolls should be transitioning into the non-linear phase. For this specific case we also plot one test run with MFV to see the differences in the early stages.

Turning then to the results, we can take from figure 5.5 that MFM and MFV both behave as expected, neatly forming the rolls with very little diffusion at a level comparable to that of mesh-based codes. If we look toward the center of the curls we can see that MFV fares slightly better, as for it we observe sharper edges with a little less diffusion than for MFM. The SPH implementations do surprisingly well with just 40 neighbors already, as unlike Hopkins we find that even with this number the KH rolls are apparent. In fact, we observe little improvement when utilizing a higher order kernel even when run at a significantly higher number of neighbors as this seemed to just very slightly decrease the amount of diffusion toward the center of the swirls.

Turning then to figure 5.6, we note immediately for t=4.2 that the issues SPH can have with this test become more relevant in the later stages, as both the quintic kernel at 40 and the Wendland kernel at 200 neighbors are showing large diffuse areas at the center of the rolls. We observe that the MFM sub-structure is smoothed out a little but is conserved well overall for t=4.2. This is also the case for the very late stages when the rolls overlap at t=8.4, where we can still observe that the characteristic swirls are present. This is not the case for the SPH codes, as we loose much of the swirls substructure already for t=4.2 where the resulting structure is more reminiscent of braids than of seperate swirls. This is the case both for the quintic and the Wendland kernel.

As we do not have an analytical solutions for this test, it is hard to ascertain which of the codes performs the best here for the very late stages at t=8.4. Nonetheless, as for the earlier ones we can clearly see MFM preserving the sub-structure better than the more diffuse SPH codes, we can therefore assume that this is the case for the later stages aswell, especially as we can still observe distinct swirls for MFM whereas for both SPH codes it would require some imagination.

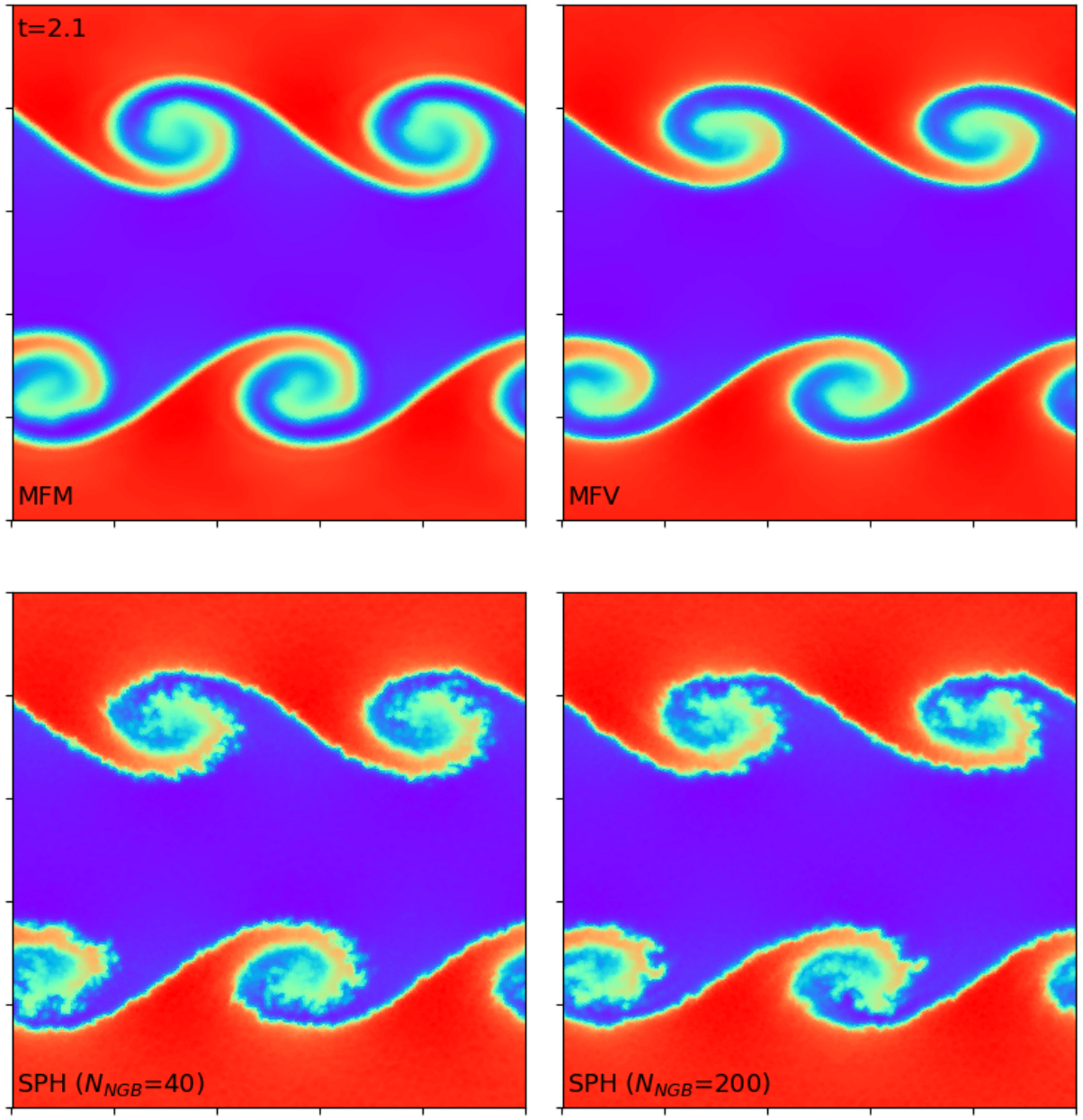



Figure 5.5: Plotted are the results for the two-dimensional KHI test run at $256^2$ resolution and plotted for t=2.1. Our MFM implementation is plotted in the *top left*, which just as MFV in the *top right* shows overall good resolution of the rolls and low amounts of noise. This becomes evident when compared to the SPH implementation run with a quintic kernel with 40 neighbors on the *bottom left*. Here we see visibly more noise present overall, which we could not improve up to the standards of MFM/MFV even when run with a Wendland kernel with 200 neighbors, here seen plotted on the *bottom right*.

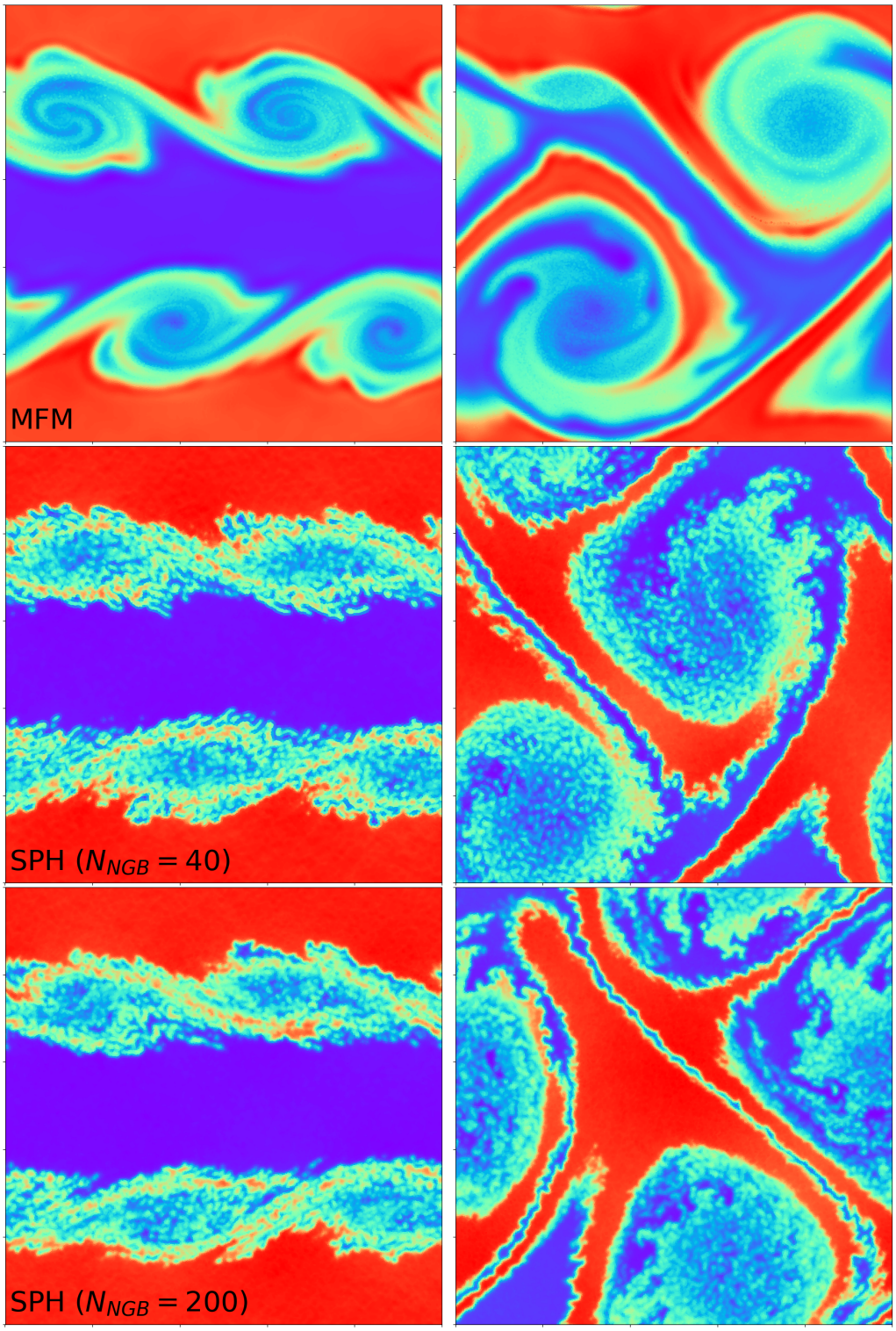



Figure 5.6: Here we see the later stages of the above test plotted. The *left column* shows the time t=4.2, whilst the *right column* shows t=8.4, again at resolution $256^2$. We can observe the transition to box-wide non-linearity as the rolls overlap with each other and where the codes start to show stark differences amongst themselves.

## 5.5 The blob test

As mentioned previously, the blob test relies on Kelvin-Helmholtz- and Rayleigh-Taylor-instabilities to tear away at the blob and finally distort it, so we expect to observe SPH having the same difficulties in capturing the evolution of the system as accurately as is possible with MFM, finding slightly more diffusion preventing the cloud from mixing as well and as quickly.

If we look at figure 5.7, we see the statement of Hopkins, namely that while SPH can have its early-time behavior agree well with MFM and mesh-based methods as our implemented version, PSPH, allows mixing in density, this is not the case for later times [8], mirrored here well. When we look at the evolution of the central blob for SPH, we can tell that that it manages to persist for much longer times than it does for MFM, with noticeable regions of upward of 4 times the ambiens density remaining intact at t=6.0 still. We also find MFM forming a similar stingray form for the blob at the time t=2.0 comparable to that found by Hopkins, with large parts already completely mixing at t=4.0. The form of the tails closely resemble each for SPH and MFM.

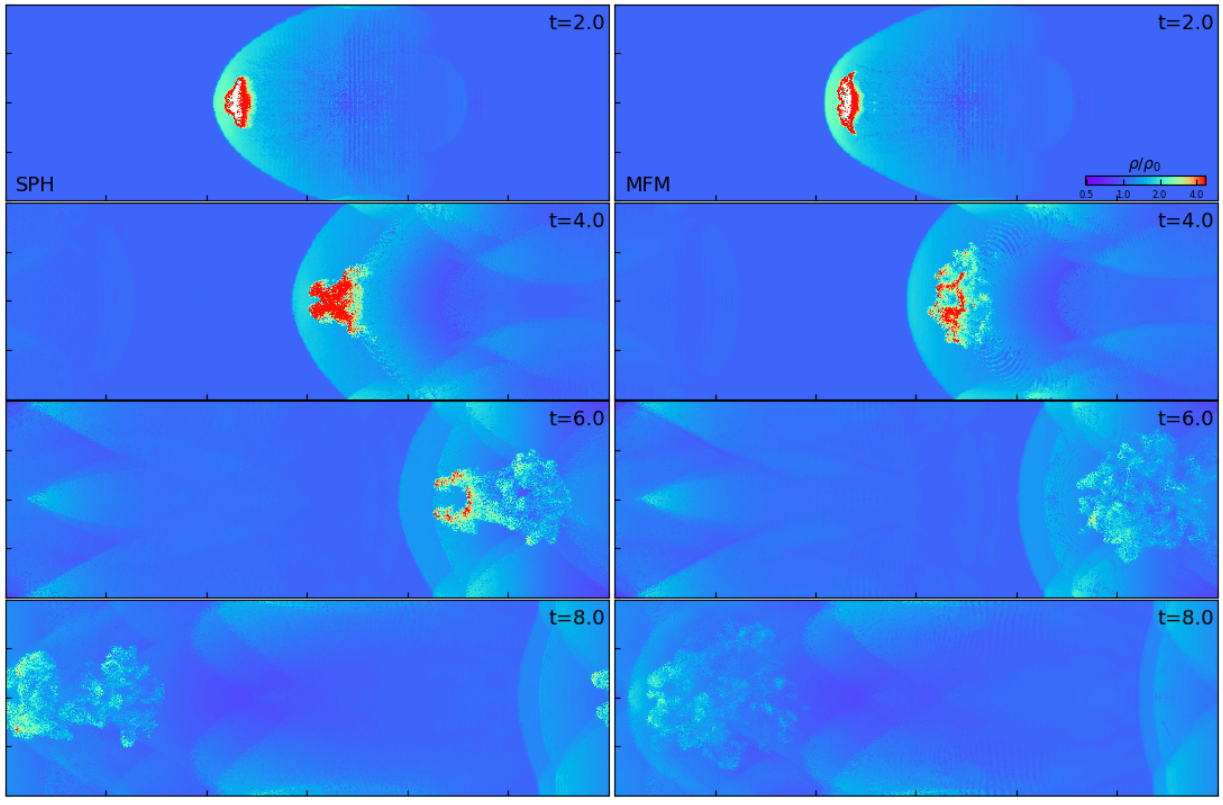



Figure 5.7: Plotted is the density in the box for a slice $(0 \leq y \leq 2000, 0 \leq z \leq 6000, x = 0)$ through the center. The ambient medium is moving to the right. The *left column* shows our results for SPH, while the *right column* shows those for MFM.

# Chapter 6

# Summary and Conclusion

Finally then we can look back at our tests and conclude our findings. Overall we observe good agreement with the results presented by Phil Hopkins, accounting for general smaller errors. For most tests we do not manage to reach the same level of optimization, as for example especially with the Sedov blast we find a significant dampening of the peak, whilst on the opposite spectrum for the KHI test we find a good forming of rolls with the SPH code for much lower numbers of neighbors than Hopkins states are necessary.

We must conclude overall though that, accounting for errors and less optimisation for our runs, MFM does indeed seem to produce consistently promising results for all considered test cases. Starting with the Gresho vortex we see excellent conservation of the peak which, while we do not manage to reproduce the exact results of Hopkins with around 90% of the original azimuthal velocity still present in the peak, offers a significant improvement over the SPH results by improving mixing and especially avoiding the E0 error. We do observe stronger diffusivity for MFM than mesh-based codes would have, however. Looking at the Sod shock tube even at our low resolution of only 320 particles, both MFM and SPH capture the converged solution well. SPH did however produce slightly larger bumps and a more prominent 'wiggle' than MFM. If we then consider the Sedov blast, we see a noteable divergence to the results of Hopkins, as we did not manage to reach the same density jump factor for MFM of 3.5, but rather find something more akin to his results for the grid codes with a jump factor of 2.72. Instead it is SPH for which we find the higher jump factor of 2.77. In return, SPH generated noticeably more noise.

Turning then to the KHI test we find excellent results for MFM and MFV, both producing very good symmetric rolls without much diffusivity (although they do not quite reach the level of a mesh-based code for this test). This is the case even for low resolution, although we then do observe noticeable smearing for the later times. The SPH codes fared remarkably well even for lower-order kernels and less neighbors, but still produced more diffuse rolls already during the early times. This is only increased for later times, where we find large differences compared to the MFM results. And finally we find quick mixing capabilities for MFM in the blob test, which are significantly faster than our used SPH version.

Summarizing then, we can state that for our tests MFM functioned as expected. It

captures shocks well as it is using a Riemann solver, and can handle mixing in the Kelvin-Helmholtz tests without the carbuncle instabilities inherent in fixed mesh codes, whilst also avoiding the fluid mixing instabilities and E0 error present in SPH. This helps it produce consistently good results. It does not seem to struggle significantly with any of our considered tests (unlike SPH with the Gresho vortex).

We conclude that MFM presents a viable alternative to the prominent codes at the time. The implementation was found to be surprisingly simple to use and gave good results even without much optimization of the parameters. We believe that it is worth exploring and further analyzing and improving its capabilities. As we could not test its behavior with tests involving self-gravity or gravity in general this could present an interesting future topic to be probed more in-depth, as this could possibly be the area where MFM has a significant advantage over mesh-based codes due to the simplicity of implementing N-body gravitation codes.

# Appendix

Here we will give the full deriavation of the energy conservation equation that forms the final of our governing equations. We will follow the derivation presented by Landau and Lifshitz in *Fluid Mechanics* [13].

The derivation begins as already alluded to in chapter 2 by looking at the expanded form of the total energy per volume element and asking how it evolves with time

$$E \equiv \tfrac{1}{2}\rho v^2 + \rho e \tag{I.1}$$

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial t}\left(\tfrac{1}{2}\rho v^2 + \rho e\right) = \frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2) + \frac{\partial}{\partial t}(\rho e) \tag{I.2}$$

We will start with the first summand, which we first expand into

$$\frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2) = \tfrac{1}{2}v^2\frac{\partial \rho}{\partial t} + \tfrac{1}{2}2\mathbf{v}\frac{\partial \mathbf{v}}{\partial t} \tag{I.3}$$

We then use the equation of continuity 2.4 for the first term and the equation of motion 2.8 for the latter and get

$$\frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2) = -\tfrac{1}{2}v^2\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho\mathbf{v}\cdot(\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \mathbf{v}\cdot\boldsymbol{\nabla}p \tag{I.4}$$

Turning then to the thermodynamic relation for the heat functional per mass we find $\mathrm{d}w = T\mathrm{d}s + \frac{V}{m}\mathrm{d}p = T\mathrm{d}s + \frac{1}{\rho}\mathrm{d}p$, which then leads us to $\boldsymbol{\nabla}p = \rho\boldsymbol{\nabla}w - \rho T\boldsymbol{\nabla}s$. We also replace $\mathbf{v}\cdot(\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} = \mathbf{v}\cdot\boldsymbol{\nabla}\left(\frac{1}{2}v^2\right)$ to arrive at the formula for our first summand

$$\frac{\partial}{\partial t}(\tfrac{1}{2}\rho v^2) = -\tfrac{1}{2}v^2\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho\mathbf{v}\cdot\boldsymbol{\nabla}\left(\tfrac{1}{2}v^2 + w\right) + \rho T\mathbf{v}\cdot\boldsymbol{\nabla}s \tag{I.5}$$

Utilizing a thermodynamic relation again, this time for the total differential of the specific internal energy, and observing that $V = \frac{m}{\rho}$ implies $\mathrm{d}V = \frac{\partial V}{\partial \rho}\mathrm{d}\rho = -m\frac{1}{\rho^2}\mathrm{d}\rho$, we get

$$\mathrm{d}e = T\mathrm{d}s - \frac{p}{m}\mathrm{d}V = T\mathrm{d}s + \frac{p}{\rho^2}\mathrm{d}\rho \tag{I.6}$$

Looking again at the heat functional per mass as defined above we observe

$$\mathrm{d}w = T\mathrm{d}s + \frac{1}{\rho}\mathrm{d}p = T\mathrm{d}s + \frac{p}{m}\mathrm{d}V - \frac{p}{m}\mathrm{d}V + \frac{V}{m}\mathrm{d}p = \mathrm{d}e - \frac{1}{m}\mathrm{d}(pV) \tag{I.7}$$

From this we can deduce $w = e + pV/m = e + p/\rho$, which we then use along with equation I.6 to solve the total differential of $\rho e$ as

$$\mathrm{d}(\rho e) = e\mathrm{d}\rho + \rho \mathrm{d}e = e\mathrm{d}\rho + \rho\left(T\mathrm{d}s + \frac{p}{\rho^2}\mathrm{d}\rho\right) = \left(e + \frac{p}{\rho}\right)\mathrm{d}\rho + \rho T\mathrm{d}s = w\mathrm{d}\rho + \rho T\mathrm{d}s \quad \text{(I.8)}$$

Using this in the second summand from equation I.2, plugging in the continuity equation again and by remembering that the specific entropy is conserved $\frac{\mathrm{d}}{\mathrm{d}t}s = \frac{\partial}{\partial t}s + \mathbf{v}\cdot\boldsymbol{\nabla}s = 0$, we arrive at

$$\frac{\partial}{\partial t}(\rho e) = w\frac{\partial \rho}{\partial t} + \rho T\frac{\partial s}{\partial t} = -w\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho T\mathbf{v}\cdot\boldsymbol{\nabla}s \quad \text{(I.9)}$$

This leads us finally to our total change in specific energy per volume, which we calculate to be

$$\frac{\partial E}{\partial t} = -\tfrac{1}{2}v^2\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho\mathbf{v}\cdot\boldsymbol{\nabla}\left(\tfrac{1}{2}v^2 + w\right) + \rho T\mathbf{v}\cdot\boldsymbol{\nabla}s - w\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho T\mathbf{v}\cdot\boldsymbol{\nabla}s \quad \text{(I.10)}$$

$$\frac{\partial E}{\partial t} = -\left(\tfrac{1}{2}v^2 + w\right)\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) - \rho\mathbf{v}\cdot\boldsymbol{\nabla}\left(\tfrac{1}{2}v^2 + w\right) \quad \text{(I.11)}$$

$$\frac{\partial E}{\partial t} = -\boldsymbol{\nabla}\cdot\left[\rho\mathbf{v}\left(\tfrac{1}{2}v^2 + w\right)\right] \quad \text{(I.12)}$$

This is then the final equation we presented in equation 2.21.

# Bibliography


[1] John D. Anderson, Jr.; *Fundamentals of Aerodynamics*, 3rd Edition, Mc-Graw Hill, 2001

[2] Oscar Agertz, Ben Moore et al.; *Fundamental differences between SPH and grid methods*, MNRAS, Volume 380, Issue 3, pp. 963-978; arXiv:astro-ph/0610051

[3] Phillip Colella; *Multidimensional Upwind Methods for Hyperbolic Conservation Laws*, Journal of computational physics Volume 87, pages 171-200, 1990

[4] Walter Dehnen, Hossam Aly; *Improving convergence in smoothed particle hydrodynamics simulations without pairing instability*, MNRAS, Volume 425, Issue 2, pp. 1068-1082.

[5] C. P. Dullemond; *Numerical Fluid Dynamics*, Online lecture on modern numerical algorithms chapter 6, 2008, `http://www2.mpia-hd.mpg.de/~dullemon/lectures/fluiddynamics08/chap_6_numhyd_riemann_1.pdf`, accessed on 17.05.2018

[6] Evghenii Gaburov, Keigo Nitadori; *Astrophysical weighted particle magnetohydrodynamics*, MNRAS, Volume 414, Issue 1, 11 June 2011, pp. 129-154

[7] RB Canelas; *SPH formulation*, online at the github wiki at `https://github.com/dualsphysics/DualSPHysics/wiki/3.-SPH-formulation`, accessed on 07.07.2018

[8] Philip F. Hopkins; *A new class of accurate, mesh-free hydrodynamic simulation methods*, MNRAS, Volume 450, Issue 1, pp.53-110, 2015; arXiv:1409.7395v2

[9] P. M. Gresho, Stevens Chan; *On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. II - Implementation*, Journal for Numerical Methods in Fluids, Volume 11, pp. 621-659, 1990

[10] J. F. Hawley, L. L. Smarr, J. R. Wilson; *A numerical study of nonspherical black hole accretion. I Equations and test problems*, AJ, Vol 277, pp. 296-311, 1984

[11] Shu-Ichiro Inutsuka; *Reformulation of Smoothed Particle Hydrodynamics with Riemann Solver*, Journal of Computational Physics, Volume 179, Issue 1, pp. 238-267, 2002

[12] Pijush Kundu, Ira Cohen, David Dowling; *Fluid Mechanics*, 6th Edition, Academic Press, 2015

[13] L. D. Landau, E. M. Lifshitz; *Fluid Mechanics*, 2nd Edition, Pergamon Press, 1987

[14] Nathalie Lanson, Jean-Paul Vila; *Renormalized Meshfree Schemes I: Consistency, Stability, and Hybrid Methods for Conservation Laws*, SIAM J. Numer. Anal., Volume 46(4), pp. 19121934, 2008

[15] D. Molteni, C. Bilello; *Riemann solver in SPH*, MmSAI, v.1., p. 36, 2003

[16] JJ Monaghan, RA Gingold; *Smoothed particle hydrodynamics theory and application to non-spherical stars.*, MNRAS, Volume 181, pp. 375389, 1977

[17] JJ Monaghan, JC Lattanzio; *A refined particle method for astrophysical problems*, A&AS, Volume 149, Issue 1, pp. 135-143, 1985

[18] JJ Monaghan; *Smoothed particle hydrodynamics*, Annual Reviews in Astronomy and Astrophysics, Volume 30, pp. 543-574, 1992

[19] JJ Monaghan; *Smoothed particle hydrodynamics*, Reports on Progress in Physics, Volume 68, Issue 8, pp.1703-1759, 2005

[20] Ue-Li Pen; *A High-Resolution Adaptive Moving Mesh Hydrodynamic Algorithm*, ApJS, Volume 115, pp. 19-34, 1998

[21] J. I. Read, T. Hayfield, O. Agertz; *Resolving mixing in smoothed particle hydrodynamics*, MNRAS, Volume 405, Issue 3, pp. 1513-1530, 2010

[22] Marco Spaans, Joseph Silk; *The Polytropic Equation of State of Interstellar Gas Clouds*, ApJS, Volume 538, Issue 1, pp. 115-120

[23] Volker Springel, Lars Hernquist; *Cosmological smoothed particle hydrodynamics simulations: The entropy equation*, MNRAS, Volume 333, pp. 649-664, 2002; arXiv:astro-ph/0111016

[24] Volker Springel; *The cosmological simulation code GADGET-2*, MNRAS, Volume 364, Issue 4, pp. 1105-1134, 2005

[25] Volker Springel; *Galileaninvariant cosmological hydrodynamical simulations on a moving mesh*, MNRAS, Volume 401, pp. 791-851, 2010

[26] J. M. Stone, T.A. Gardiner, P. Teuben, J. F. Hawley, J. B. Simon; *Athena: A New Code for Astrophysical MHD*, ApJS, Volume 178, pp. 137-177, 2008; arXiv:0804.0402

[27] Eleuterio F. Toro; *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 1st Edition, Springer, 1997

[28] Eleuterio F. Toro; *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd Edition, Springer, 2009

[29] Bram van Leer; *On the Relation Between the Upwind-Differencing Schemes of Godunov, EngquistOsher and Roe*, SIAM Journal on Scientific and Statistical Computing, Volume 5, No. 1 : pp. 1-20, 1984

[30] Damien Violeau; *Fluid Mechanics and the SPH Method: Theory and Applications*, 1st Edition, Oxford University Press, 2012

[31] Holger Wendland; *Piecewise polynomial, positive definite amd compactly supported radial functions of minimal degree*, Adv. Comp. Math., Volume 4, Issue 1, pp. 389-396, 1995

[32] Gouhong Xu; *Hydrodynamic and N-body schemes on an unstructured, adaptive mesh with applications to cosmological simulations*, MNRAS Volume 288, pp. 903-919, 1997

[33] Ya. B. Zel'dovich; *Gravitational instability: An approximate theory for large density perturbations.*, A&AS, Volume 5, pp. 84-89, 1970

# Acknowledgments

The usage of first person plural throughout this thesis is not by convention only, as without the following people this work would not have been possible. I would therefore like to take this moment to thank all these wonderful people who directly or indirectly guided me through this specific work and the entire three years of the bachelor itself.

First and foremost I would like to thank my supervisors, Dr. Benjamin Moster and Ulrich Steinwandel, for the possibility of diving into this fascinating topic and their consistently excellent support of my endeavours, for all the late-night email answers and the great explanations that often went beyond the scope of what I could present here. Thank you for helping me break down and understand many of the underlying concepts in great depth and with a lot of patience. I would also like to thank Joseph O'Leary for his outstanding assistance in all things related to the plotting of the tests.

Second, I would like to thank the wonderful people surrounding me, most prominently my family and good friends, both inside and outside of germany. My family for their support (and of course funding) of my studies, constant love and compassion and a lot of nerves when dealing with my panicked calls before important due dates. My friends for listening to my speaking of 'interesting physical topics' that really just amounted to good excuses to hold a monologue about how splendid physics and maths are, for their constant build-ups and great conversations to take my mind off of (physics-related) things. I hold all of you very dearly.

Then I would like to explicitly thank Robin Evitts and Stacey Kimmig for thoroughly proofreading all the above pages, so any grammar or spelling complaints may be directed at them. In all honesty however, their feedback significantly improved the readability of this thesis, and for that thank you.

Then finally I would like to thank all my fellow physics students for all the great times and memories, especially those with whom I spent large amounts of time playing board games and studying for exams with, all the people at the USM for their kindness and willingness to help whenever necessary and finally once more all the mentioned people above. Thank you.

# Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbstständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben:

München, den _____ _____

(Lucas Kimmig)