CES-Seminar WS 13/14

Smoothed Particle Hydrodynamics

Author: Tony Rosemann
RWTH Aachen University

Supervisor: Prof. Dr. Manuel Torrilhon
Center for Computational Engineering Science
Mathematics Division
RWTH Aachen University

Contents

1 Introduction

Smoothed particle hydrodynamics (SPH) was developed in 1977 by Monaghan and Gingold [7] and Lucy [3] independently to solve complex astrophysical problems governed by fluid equations. They needed a numerical method which was able to achieve reasonable accuracy with a small number of discretization points. Furthermore, they wanted this method to be easy to implement and robust. However, the finite-difference techniques available at that time did not meet these requirements. That was their motivation to design the new method in which spatial derivatives are not approximated by using fixed vertices on a mesh, like in the finite-difference schemes. In SPH one uses particles that move with the flow and spatial derivatives or other quantities are estimated by kernel interpolation between the particles. Although SPH was originally designed to investigate astrophysical phenomena, it has proven to be an useful method for solving many other fluid dynamical problems. How this method works and how it can be applied to fluid equations will be discussed in this paper.
Chapter 2 outlines the mathematical essentials of SPH and chapter 3 deals with the SPH version of the Euler equations. Those two chapters basically rely on a review by Monaghan [6]. Where other references have been used, these are specifically given in the text. In chapter 4 the results of a SPH simulation are presented.

2 Mathematical aspects of SPH

2.1 Interpolation and particle summation

SPH is based on the interpolation of function values. Consider an arbitrary function $f(\vec{r})$. The value of f at position $\vec{r}$ can be interpolated by the integral interpolant

$$\tilde{f}(\vec{r}) = \int f(\vec{r'}) W(|\vec{r} - \vec{r'}|, h) d\vec{r'}, \tag{1}$$

where W is called interpolation kernel or smoothing function which is usually shaped like a Gaussian distribution. The parameter h determines the kernel's width and is referred to as smoothing length. This interpolation is exact if W is the Dirac delta function δ. Thus, one chooses W such that it tends to δ as h tends to zero:

$$\lim_{h \to 0} W(|\vec{r} - \vec{r'}|, h) = \delta(|\vec{r} - \vec{r'}|). \tag{2}$$

Another property of W is its normalization:

$$\int W(|\vec{r} - \vec{r'}|, h) d\vec{r'} = 1. \tag{3}$$

As a consequence, the interpolation error for constant functions is zero.
Equation (1) can be rewritten as

$$\tilde{f}(\vec{r}) = \int \frac{f(\vec{r'})}{\rho(\vec{r'})} W(|\vec{r} - \vec{r'}|, h)\rho(\vec{r'})d\vec{r'}, \tag{4}$$

where ρ denotes the density. Note that $\rho d\vec{r'}$ is an infinitely small mass element. The continuum is now replaced by a finite number of point masses, referred to as particles with mass m_b and density ρ_b. This set of particles represents the fluid in SPH. As a result, the integral interpolant becomes the summation interpolant

$$f(\vec{r}) = \sum_b m_b \frac{f_b}{\rho_b} W(|\vec{r} - \vec{r}_b|, h), \tag{5}$$

where f_b denotes the value of f at the particle's position $\vec{r}_b$. The local density of a fluid can therefore be estimated by the summation of weighted particle masses:

$$\rho(\vec{r}) = \sum_b m_b W(|\vec{r} - \vec{r}_b|, h). \tag{6}$$

For many applications it is sufficient to use this approximation in a SPH simulation instead of solving the continuity equation.

2.2 Kernels

Monaghan and Gingold used in their first SPH simulation the Gaussian kernel

$$W(r, h) = \frac{\sigma}{h^d} \exp(-q^2), \tag{7}$$

where d is equal to the number of dimensions, $r = |\vec{r} - \vec{r'}|$ and $q = r/h$. The normalization constant σ is given by $[1/\sqrt{\pi}, 1/\pi, 1/(\sqrt{\pi}\pi)]$ in one, two and three dimensions, respectively. The disadvantage of this kernel is that it has no compact support. Thus, one has to take into account all particles in the summation interpolant, which is computationally inefficient, because the influence of particles far away is negligible. That is why SPH simulations usually use M_n splines which are still Gaussian shaped, but have a compact support and continuous derivatives up to order $(n-2)$. The most commonly used M_4 spline is given by

$$M_4(r, h) = \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & 0 \leq q \leq 1 \\ \frac{1}{4}(2-q)^3, & 1 < q \leq 2 \\ 0, & q > 2 \end{cases} \tag{8}$$

and the kernel based on this cubic spline reads

$$W(r,h) = \frac{\sigma}{h^d} M_4(r,h) \tag{9}$$

with $\sigma = [2/3, 10/(7\pi), 1/\pi]$. In a simulation with N particles the numerical effort to calculate a summation interpolant for each particle reduces from $\mathcal{O}(N^2)$ to $\mathcal{O}(nN)$ when using the spline-based kernel instead of the Gaussian. The number of particles within the support of W denoted by n can be determined by using appropriate neighbour search algorithms. The two smoothing functions presented here are shown in figure 1.

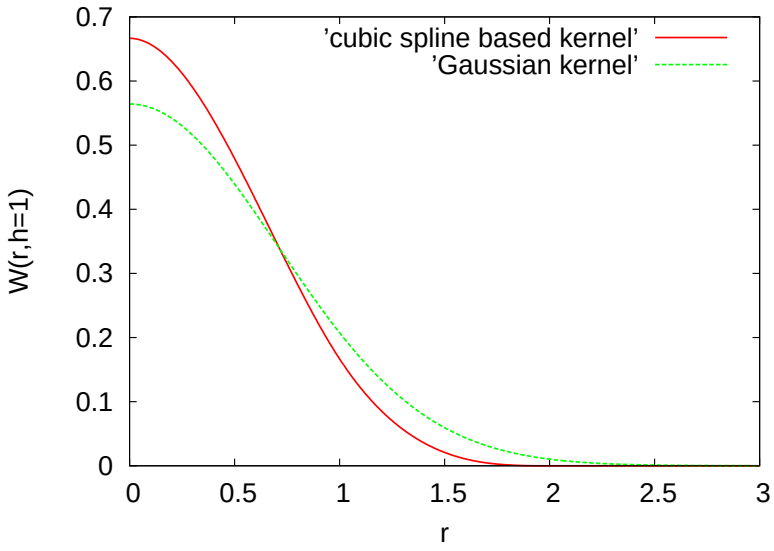


Figure 1: Gaussian kernel and cubic spline-based kernel for $d = 1$ and $h = 1$. The spline-based kernel has a compact support of $2h$, whereas the Gaussian has no compact support.

Since the particle density can vary in space and time, the smoothing length h should also vary to obtain reasonable interpolation results. To change $h_a = h(\vec{r}_a)$ according to

$$h_a = k \left(\frac{m_a}{\rho_a} \right)^{1/d} \tag{10}$$

with $k \sim 1.4$ has been proven to be a robust and simple way to specify h. Because the resolution can be easily adjusted with this formula and the kernel can also be exchanged without much effort, SPH codes are very flexible.

2.3 Derivatives in SPH

Since the purpose of the method is to solve fluid equations, it is necessary to estimate spatial derivatives. But this turns out to be very easy within the SPH framework. If W is a differentiable function, the gradient of the summation interpolant (5) at the position of particle a

reads

$$\nabla f_a = \sum_b m_b \frac{f_b}{\rho_b} \nabla_a W_{ab}, \tag{11}$$

where W_{ab} denotes $W(|\vec{r}_a - \vec{r}_b|, h)$ and ∇_a the gradient taken with respect to the coordinates of particle a. Because W is a known function, this means that one can calculate the exact derivatives of interpolants in SPH and that there is no need no use finite-difference methods. The gradient of W can be written as

$$\begin{aligned} \nabla_a W_{ab} &= \nabla_a W(|\vec{r}_a - \vec{r}_b|, h) \\ &= \frac{\partial}{\partial \vec{r}_a} W(|\vec{r}_a - \vec{r}_b|, h) \\ &= \frac{\partial W(r_{ab}, h)}{\partial r_{ab}} \frac{\partial |\vec{r}_a - \vec{r}_b|}{\partial \vec{r}_a} \\ &= \frac{\partial W(r_{ab}, h)}{\partial r_{ab}} \hat{e}_{ab}, \end{aligned} \tag{12}$$

where r_{ab} denotes specifically $|\vec{r}_a - \vec{r}_b|$ and $\hat{e}_{ab} = (\vec{r}_a - \vec{r}_b)/r_{ab}$ is the unit vector pointing from particle b to particle a.

Unfortunately the gradient approximation provided by equation (11) does not vanish for constant function values ($f_a = f_b$). But this can be achieved by using the product rule

$$\nabla f = \frac{1}{\Phi} \left(\nabla(\Phi f) - f \nabla \Phi \right), \tag{13}$$

where Φ is a differentiable function. For example, choosing $\Phi = \rho$ and evaluating the two gradients in this expression with equation (11) gives the gradient approximation

$$\begin{aligned} \nabla f_a &= \frac{1}{\rho_a} \left(\sum_b m_b \frac{(\rho f)_b}{\rho_b} \nabla_a W_{ab} - f_a \sum_b m_b \frac{\rho_b}{\rho_b} \nabla_a W_{ab} \right) \\ &= \frac{1}{\rho_a} \sum_b m_b (f_b - f_a) \nabla_a W_{ab}, \end{aligned} \tag{14}$$

which vanishes if f is constant.

Curl and divergence can be calculated in the same way. Applying the operators to the summation interpolant of a vector-valued function $\vec{f}$ and using again equation (13) with $\Phi = \rho$ gives

$$(\nabla \cdot \vec{f})_a = \frac{1}{\rho_a} \sum_b m_b (\vec{f}_b - \vec{f}_a) \cdot \nabla_a W_{ab} \tag{15}$$

and

$$(\nabla \times \vec{f})_a = \frac{1}{\rho_a} \sum_b m_b (\vec{f}_b - \vec{f}_a) \times \nabla_a W_{ab}. \tag{16}$$

In principle second order derivatives could be obtained by simply differentiating the kernel twice. But this approach is disadvantageous, because these equations would be very sensitive to particle disorder and the kernel's second order derivative is not positive definite. The sign of the estimated derivative would, therefore, depend on the particle separation, which is not desirable. For example, when applying SPH to the heat equation, hot particles should always transfer heat to colder ones no matter how far they are separated from each other. But it is possible to calculate estimates for second order derivatives which are still based on the kernel's first derivative. More details on this issue can be found in Monaghan's review [6].

2.4 Errors

The numerical error of the integral interpolant can be calculated by using the taylor series of f. For an even smoothing function one can show

$$\tilde{f}(\vec{r}) = f(\vec{r}) + c(W, f^{(2)}(\vec{r})) \cdot h^2 + \mathcal{O}(h^4), \tag{17}$$

where c is a quantity which includes a constant depending on the kernel and the second order derivatives of f. First and third order terms vanish because of the kernel's symmetry. As a result, the integral interpolant provides a second order interpolation. Note that this error estimation is not valid at the boundary, because the whole support of W is not within the domain. That is why higher errors can be expected in the boundary region. It is also possible to design kernels, where c equals zero, leading to a fourth order interpolation. But these kernels change necessarily sign, which is not desirable. It would then be possible to obtain a negative value for the density interpolant near strong shocks. However, techniques have been developed using a switch from high order to low order kernels near shocks. But this has not been fully explored.

For the summation interpolant one can also show a second order error, but only when the particles are assumed to be ordered. Because this is generally not the case due to the dynamics of the system, there is usually no classical error estimation for the results of a SPH simulation. Monaghan and Gingold expected the error of their first SPH simulation to be $\sim 1/\sqrt{N}$, like one would expect it in Monte Carlo simulations, another particle method. They were surprised that the actual error was much lower than this estimate. This can be observed in many other SPH simulation, too. The reason for smaller errors is that the SPH approximations based on smoothing can handle fluctuations which are not consistent with the true dynamics. As a consequence, particles are kind of orderly disordered.

3 SPH Euler equations

Since its development in the late 1970s, SPH has been applied to many problems including compressible and incompressible flows, multi-phase flows, free surface flows, magnetohydrodynamics, elasticity and fracture, shocks, heat conduction and it is still finding more applications. This chapter shows how the method can be applied to a simple set of fluid equations. The Euler equations for the density ρ, the velocity $\vec{v}$ and the position $\vec{r}$ are given by

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \vec{v} \tag{18}$$

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho}\nabla P \tag{19}$$

$$\frac{d\vec{r}}{dt} = \vec{v}, \tag{20}$$

where P denotes the pressure and gravitational forces have been neglected. Equation (18) is also referred to as continuity equation. In the absence of any entropy generation one can derive from the first law of thermodynamics the time rate change of internal energy per unit mass u:

$$\frac{du}{dt} = \frac{P}{\rho^2}\frac{d\rho}{dt}. \tag{21}$$

The equations can be closed with suitable equation of state. For ideal gases one may use

$$P = \rho u(\gamma - 1), \tag{22}$$

where γ denotes the adiabatic index.
It should be clear that particles change their position according to their current velocity (equation 20) and their pressure values can be calculated from their internal energy and density values (equation 22). The SPH version of continuity equation (18), acelareation equation (19) and energy equation (21) will be derived and discussed below.

3.1 Continuity equation

Remember equation (15) from chapter 2.3 which provided an estimate for the divergence of a vector-valued function. Using this interpolant to approximate $\nabla \cdot \vec{v}$ in the continuity equation one finds

$$\frac{d\rho_a}{dt} = \sum_b m_b(\vec{v}_a - \vec{v}_b) \cdot \nabla_a W_{ab}. \tag{23}$$

There are certain cases in which it is advantageous to use this equation, but as already mentioned in chapter 2, almost all SPH simulations use the interpolant

$$\rho_a = \sum_b m_b W_{ab}. \tag{24}$$

In both cases mass conservation is exact, because the particles' masses are generally fixed in a SPH simulation.

3.2 Acceleration equation

If the pressure gradient in the acceleration equation (19) is estimated by interpolant (14), one finds the SPH version

$$\frac{d\vec{v}_a}{dt} = \frac{1}{\rho_a^2} \sum_b m_b (P_a - P_b) \nabla_a W_{ab}. \tag{25}$$

The disadvantage of this interpolant is that it does not conserve linear and angular momentum. To illustrate that, one can consider particle a and another particle b. The force on particle a owing to particle b is then given by

$$\vec{F}_a = \frac{1}{\rho_a^2} m_a m_b (P_a - P_b) \nabla_a W_{ab} \tag{26}$$

and the force on particle b owing to particle a by

$$\vec{F}_b = -\frac{1}{\rho_b^2} m_a m_b (P_a - P_b) \nabla_b W_{ba}. \tag{27}$$

Obviously, the two forces are not equal and opposite:

$$\vec{F}_a \neq -\vec{F}_b. \tag{28}$$

Thus, Newton's third law is not fulfilled and total momentum is not conserved. However, a momentum conserving approximation can be obtained using

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho. \tag{29}$$

With the help of this expression the interpolant for the acceleration equation can be rewritten as

$$\frac{d\vec{v}_a}{dt} = -\sum_b m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} \right) \nabla_a W_{ab}. \tag{30}$$

Because of the equation's symmetry, the forces are now equal and opposite if

$$\nabla_a W_{ab} = -\nabla_b W_{ba}. \tag{31}$$

This equality holds for equal smoothing lengths, but it is generally not given if h is allowed to vary, for example according to equation (10). But it is possible to maintain momentum conservation for varying smoothing lengths by symmetrizing the kernels. This can be done

by taking the arithmetic mean of the the smoothing lengths:

$$\bar{h}_{ab} = \frac{1}{2}(h_a + h_b) \tag{32}$$

and rewrite the kernel as

$$\tilde{W}_{ab} = W(|\vec{r}_a - \vec{r}_b|, \bar{h}_{ab}) \tag{33}$$

as suggested by Monaghan [5] or by taking the mean of the kernels:

$$\tilde{W}_{ab} = \frac{1}{2}(W(|\vec{r}_a - \vec{r}_b|, h_a) + W(|\vec{r}_a - \vec{r}_b|, h_b)) \tag{34}$$

as suggested by Hernquist and Katz [2].
To stabilize the numerical algorithm one can add artificial viscosity to the equation, which is especially necessary when simulating shocks. A commonly used model proposed by Monaghan [5] adds a viscous term Π_{ab} to equation (30):

$$\frac{d\vec{v}_a}{dt} = -\sum_b m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a W_{ab}, \tag{35}$$

where Π_{ab} is given by

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha \bar{c}_{ab} \mu_{ab} + \beta \mu_{ab}^2}{\bar{\rho}_{ab}} & \vec{v}_{ab} \cdot \vec{r}_{ab} < 0 \\ 0 & \vec{v}_{ab} \cdot \vec{r}_{ab} > 0 \end{cases} \tag{36}$$

and the viscosity by

$$\mu_{ab} = \frac{\bar{h}_{ab} \vec{v}_{ab} \cdot \vec{r}_{ab}}{r_{ab}^2 + \eta^2}, \tag{37}$$

with $\vec{v}_{ab} = \vec{v}_a - \vec{v}_b$ and $\vec{r}_{ab} = \vec{r}_a - \vec{r}_b$. The quantities $\bar{\rho}_{ab}$ and $\bar{c}_{ab}$ denote the average density and the average speed of sound (e.g. $c = \sqrt{\gamma P/\rho}$ for ideal gas) and are computed similar to the mean of the smoothing lengths $\bar{h}_{ab}$ given by equation (32). The parameters α, β and η are typically chosen to be $\alpha = 1.0$, $\beta = 2.0$ and $\eta = 0.1\bar{h}_{ab}$. This viscous term smoothes the velocity by inducing an additional repelling force on particles which approach other ($\vec{v}_{ab} \cdot \vec{r}_{ab} < 0$) and which are separated by a short distance r_{ab}. The term that is linear in μ_{ab} produces shear and bulk viscosity and the quadratic one handles high Mach numbers. Total linear and angular momentum are still conserved when adding this artificial viscosity.

3.3 Energy equation

The energy equation (21) can be converted into a SPH version by approximating the density's time rate change with equation (23):

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b (\vec{v}_a - \vec{v}_b) \cdot \nabla_a W_{ab}. \tag{38}$$

If the artificial viscosity (36) is used, one has to rewrite this equation as

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b (\vec{v}_a - \vec{v}_b) \cdot \nabla_a W_{ab} + \frac{1}{2} \sum_b m_b \Pi_{ab} (\vec{v}_a - \vec{v}_b) \cdot \nabla_a W_{ab} \tag{39}$$

to be consistent [8]. When equation (30) together with (38) or when equation (35) together with (39) is used, the total energy

$$E = \sum_a m_a \left(u_a + \frac{1}{2} \vec{v}_a \cdot \vec{v}_a \right) \tag{40}$$

is conserved. Of course one has to take the same symmetrized kernel in both equations to ensure conservation. A detailed proof for the momentum and energy conservation of the presented equations can be found in Rosswog [8].

3.4 Time integration

Because SPH transforms fluid equations into ordinary differential equation, one could use any time integrator of desired accuracy. However, a fourth-order Runge-Kutta method, for example, would not conserve total momentum and there are situations in which an algorithm of lower order, but conserving the momentum, delivers much better results. That is why one often prefers integrators which have good conservation properties like the Leapfrog method which provides a second-order approximation.
Another second-order integrator is a predictor-corrector method developed by Monaghan [4]. It conserves linear and angular momentum and can be used to maintain the conservation properties of the SPH equations which have been discussed in the previous chapters. Monaghan also provides a time step control for this method to ensure stability of the algorithm. It includes a rule based on the force:

$$\Delta t_f = \min_a \sqrt{\frac{h}{|\vec{a}_a|}}, \tag{41}$$

where $\vec{a}$ denotes the acceleration and it includes a rule which combines the Courant condition with stability conditions:

$$\Delta t_c = \min_a \frac{h}{c_a + 0.6(\alpha c_a + \beta \max_b \mu_{ab})}, \tag{42}$$

where μ_{ab} is given by equation (37) and the parameters α and β are the same as in equation (36). The timestep is then taken as

$$\Delta t = 0.25 \min(\Delta t_f, \Delta t_c). \tag{43}$$

4 Shock tube problem

In 1978 Gary Sod [10] analysed the behaviour of several finite-difference methods in a shock tube experiment that has become a popular numerical benchmark. In this chapter the results of a SPH shock tube simulation are presented. Sod's experiment considers a one-dimensional tube with a membrane at $x = 0$ which breaks at $t = 0$. Each side of the membrane is filled with ideal gas, where the gas on the left side is stored under higher pressure and has a higher density. The initial conditions shown in figure 2 are given by $[\rho_l, P_l, v_l, \gamma_l] = [1.0, 1.0, 0.0, 1.4]$ and $[\rho_r, P_r, v_r, \gamma_r] = [0.125, 0.1, 0.0, 1.4]$. The state equation (22) gives the initial internal energies $u_l = 2.5$ and $u_r = 2.0$.

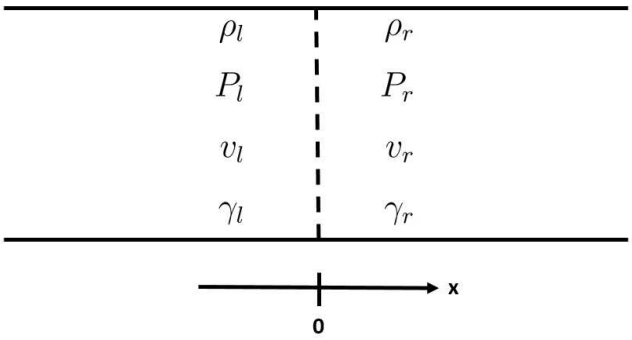


Figure 2: Initial condtions of the Sod shock tube problem.

As time evolves one will observe 5 characteristics, namely from left to right: initial condition "l", a rarefaction wave, a contact discontinuity, a shock discontinuity and initial condition "r". The system can be described by Euler's equations (18)-(20), energy equation (21) and state equation (22).

For the SPH simulation results presented here density interpolant (24), acceleration interpolant (35) the energy time rate change interpolant (39) were used. The pressure was calculated from the state equation at each time step. The artificial viscosity model and its associated parameters given in chapter 3.2 were used. All interpolants were calculated with the cubic spline kernel (9) and symmetrized with equation (34). The smoothing lengths were specified according to equation (10). For time integration the predictor-corrector method by Monaghan [4] and the time step control from chapter 3.4 were used. Since this simulation uses 405 particles with equal mass, the spacing had to be adjusted accordingly. More specifically, because the density is initially eight times higher on the left side than on the right side of the membrane, the spacing has to be eight times smaller on the left side to be consistent when using equimass particles. Thus, there were initially $8N/9$ particles on the left side and $N/9$ on the right side. The particles were placed from $x = -0.5$ to $x = 0.5$ with equidistant spacing on each side of the membrane. At the boundaries 10 additional particles with fixed initial conditions were inserted.

Figure 3 shows the velocity profile with the artificial viscosity turned off. As expected, strong oscillations can be observed, which shows that the numerical algorithm tends to become unstable when simulating shocks. But due to the conservation properties of SPH, total energy is still conserved up to under one percent.

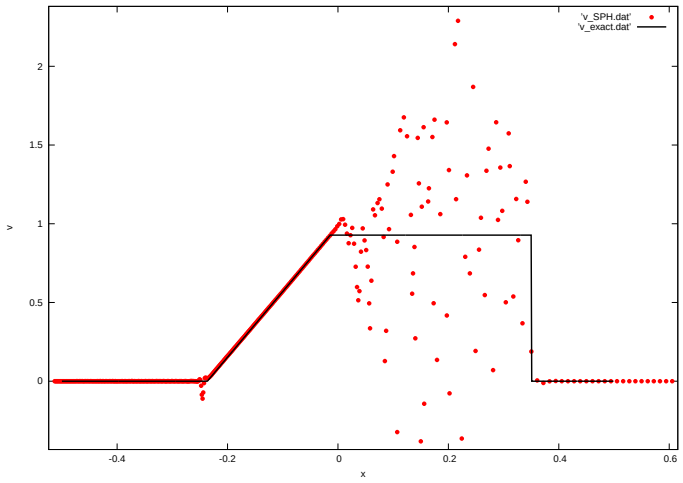
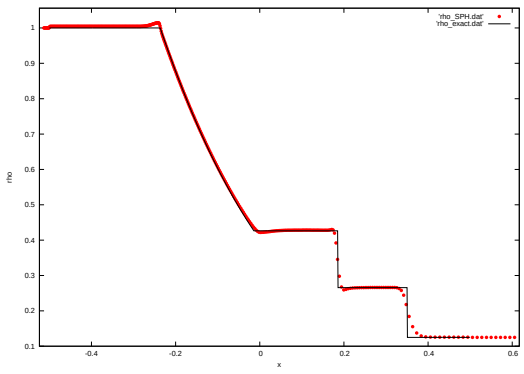
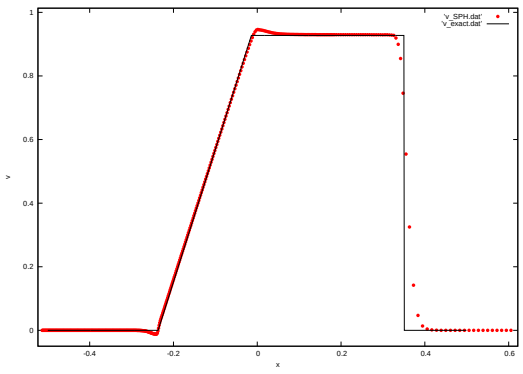


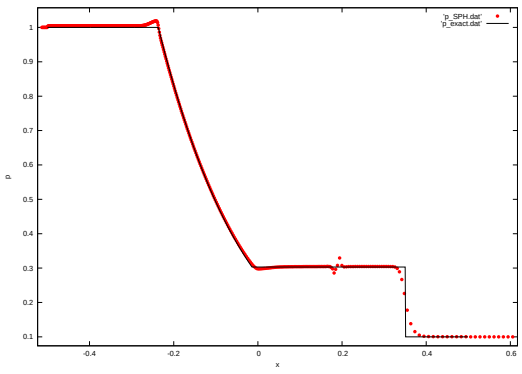
Figure 3: Exact (black line) and numerical (red dots) velocity against position for the Sod shock tube at t=0.2s. Artificial viscosity was turned off in this case.



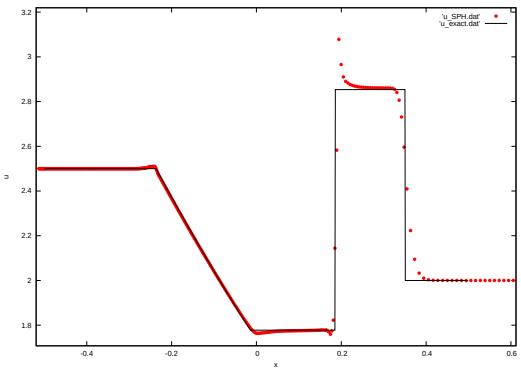
(a) Density



(b) Velocity



(c) Pressure



(d) Internal energy

Figure 4: Results of the Sod shock tube simulation at t=0.2s with artificial viscosity turned on.

Figure 4 shows that the oscillations disappear when applying artificial viscosity. Despite the small overshoots at the ends of the rarefaction wave, the smearing of the shock discontinuity and the spike in the internal energy, the SPH solution agrees very well with analytical solution taken from Toro [11]. Total energy is conserved up to $\sim 0.037\%$ in this test. Of course better results can be obtained with Riemann solvers, for instance. But the purpose of SPH is achieving reasonable, not highest, accuracy without much effort. However, it is possible to replace the artificial viscosity with exact or approximative Riemann solvers and combine their powerful shock capturing abilities with the simplicity and robustness of SPH. (e.g. Cha & Whitworth [1], Sirotkin & Yoh [9]).

5 Conclusion

This paper gave a short introduction to Smoothed particle hydrodynamics. The basic idea of SPH to divide the fluid into particles and interpolate between them is quite intuitive. Since it is gridless method, the implementation effort is in many cases relatively low and SPH codes are very flexible. It is possible to visualize the basic dynamics of fluid with a small number of particles and the simulation results guarantee conservation of mass, momentum and energy when using the right interpolation formulas. This makes SPH a very robust method.
However, if the purpose of the simulation is to obtain very high accuracy or to resolve small perturbations, finite-difference techniques are usually the better choice.

References

[1] S.-H. Cha and A. P. Whitworth. Implementations and tests of godunov-type particle hydrodynamics. *Monthly Notice of the Royal Astronomical Society*, 340:73–90, 2003.

[2] L. Hernquist and N. Katz. Treesph - a unification of sph with the hierarchical tree method. *Astrophysical Journal Supplement Series*, 70:419–446, 1989.

[3] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(12):1013–1024, 1977.

[4] J. J. Monaghan. On the problem of penetration in particle methods. *Journal of computational physics*, 82:1–15, 1989.

[5] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.

[6] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759, 2005.

[7] J. J. Monaghan and R. A. Gingold. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

[8] S. Rosswog. Astrophysical smooth particle hydrodynamics. *New Astronomy Reviews*, 53:78–104, 2009.

[9] F. V. Sirotkin and J. J. Yoh. A smoothed particle hydrodynamics method with approximate riemann solvers for simulation of strong explosions. *Computers & Fluids*, 88:418–429, 2013.

[10] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.

[11] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer-Verlag Berlin Heidelberg, 1997.