PROCEEDINGS, Thirty-Sixth Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, January 31 - February 2, 2011
SGP-TR-191

SIMULATION OF TWO-PHASE FLOW IN GEOTHERMAL PIPES USING SMOOTHED PARTICLE HYDRODYNAMICS

Heimir Hjartarson*, Magnús Þ. Jónsson and Halldór Pálsson

University of Iceland
Hjarðarhagi 2-6
Reykjavík, 107, Iceland
e-mail: heh5@hi.is

ABSTRACT

In geothermal power plants, a pipe system is used to gather fluids from production wells and transport them to a power plant, or to steam separators. In the case of hydrothermal systems, where the geothermal fluid is a mixture of steam and water, this gathering system is normally designed for two-phase flow. An important part of the design process is to determine flow conditions (or regimes) in the pipes as well as pressure variations. The smoothed particle hydrodynamics (SPH) method is a meshfree Larangian method that has recently gained increased interest in computational fluid dynamics. The method is particularly beneficial in case of complex multiphase flow. In this work, the SPH method is programmed and implemented in C++, based on recent publications in the field. A well known case of a shock tube in one-dimension is modeled with good results. The problem is extended to three-dimensions, but preliminary results indicate that treatment of boundaries needs improvement.

INTRODUCTION

During the last century, rapid development in the utilization of geothermal energy for district heating and electricity production has been ongoing in Iceland. Today there are several geothermal power plants operational and there is potential for more. Because of the nature of the geothermal reservoirs involved, the steam supply systems for these power plants are normally designed for two-phase flow (Ballzus et al, 1992).

When designing such systems it is important to model the components of the steam supply system, e.g. the pipes and steam separators, and to be able to predict pressure variations. Steam separators in particular are large and expensive pressure vessels and allowable size reduction or shape change could significantly reduce capital cost of geothermal power plant projects. This paper describes the initial work of a project, where the physical behavior of fluid inside a steam separator will be studied, using the smoothed particle hydrodynamics (SPH) method. The first step towards that goal is to simulate flow in a relatively simple geothermal pipe and the SPH method has been selected for that purpose.

The smoothed particles hydrodynamics (SPH) method was first introduced for solving astrophysical problems (Lucy, 1977;Gingold & Monaghan, 1977). Since then, the method has been expanded to other areas of engineering and science, including multi-phase flow (Liu & Liu, 2003, Monaghan 2005).

When comparing the SPH method to conventional grid-based numerical methods, like finite difference method (FDM) and finite element method (FEM), it turns out that the two latter methods are better suited for some applications (Liu & Liu, 2003) and also very well proven in the field of computational fluid dynamics (CFD). The advantage of the SPH method are present in areas where these traditional grid-based methods are hard to apply, e.g. in free surface flows, for deformable boundaries, moving interfaces and problems with large deformations of the computational domain. Other cases where SPH is well suited are complex mesh generation cases, difficult mesh adaptivity problems, and problems with multiscale resolution (Liu & Liu, 2003). According to Liu & Liu (2010) and Monaghan (2005) other attractive features of the SPH method are e.g. that pure advection is treated exactly, interface problems with more than one fluid type are relatively easy to represent (since each fluid type has it own set of particles) and the SPH method it is often easier to implement numerically, compared to grid-based methods. The method is nevertheless not problem free, since boundary representation is still a challenging problems, with different solution proposals, see Monaghan (2005).

In this paper the SPH method is introduced and a programming framework for the method is described. The SPH program is applied to a shock tube problem in one-dimension and three-dimensions, and the results presented. Finally, conclusions are drawn and future work for implementing a two-phase model is discussed.

SMOOTHED PARTICLE HYDRODYNAMICS

Fundamentals

The SPH method is a meshfree Larangian particle method where a flow field is represented by a set of particles, which have properties such as volume, density and mass. These particles then interact with each other within a range of a specific smoothing function (or kernel function) and their relevant properties are approximated as integral interpolants. The following explanation is based on work of Monaghan (1992). The integral interpolant of any function A(**r**) is defined by:

$$A_I(\bar{r}) = \int A(\bar{r}')W(\bar{r}-\bar{r}',h)d\bar{r}' \qquad (1)$$

where h is an influence radius or smoothing length, **r** is a position vector and W is a weight function. The interpolating kernel should satisfy a number of conditions. The two most important are the normalization condition, given by

$$\int W(\bar{r}-\bar{r}',h)d\bar{r}' = 1 \qquad (2)$$

and the Delta function property, which is represented as

$$\lim_{h\to 0} W(\bar{r}-\bar{r}',h) = \delta(\bar{r}-\bar{r}')$$

The interpolant is then approximated by a summation interpolant

$$A_S(\bar{r}) = \sum_b m_b \cdot \frac{A_b}{\rho_b} W(\bar{r}-\bar{r}_b,h) \qquad (3)$$

where the index b is a particle label with mass m_b, position $\bar{r}_b$, density ρ_b. Any other property of A at $\bar{r}_b$ is represented in A_b. The summation is then performed over all particles within the influence of particle b.

The gradient of function A can be obtained by simply calculating the gradient of the kernel function

$$\nabla A_S(\bar{r}) = \sum_b m_b \cdot \frac{A_b}{\rho_b} \nabla W(\bar{r}-\bar{r}_b,h) \qquad (4)$$

A good example of a kernel function is the Gaussian kernel, which was presented in the original work of Gingold & Monaghan (1977)

$$W(r,h) = \alpha_d e^{-(r^2/h^2)}$$

where r is the distance between particles and α_d is $1/\pi^{1/2}h$, $1/\pi h^2$, $1/\pi^{3/2}h^2$, respectively in one, two and three dimensions. This is necessary to satisfy the unity condition.

SPH Equations of Motion

There are number of ways to derive the SPH formulation from the governing equations, e.g. the Navier-Stokes or Euler equations for fluid flow (Liu & Liu, 2003, Monaghan, 2005). The momentum equation for the Euler then becomes

$$\frac{d\bar{v}_a}{dt} = -\sum_b m_b \cdot \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)\nabla_a W_{ab} \qquad (5)$$

where $\bar{v}_a$ is the velocity of particle a, P is pressure and the term $\nabla_a W_{ab}$ is the kernel function based on the distance $\bar{r}_b - \bar{r}_a$ and the smoothing length h. For the continuity equation there are two commonly used forms, the summation density given by

$$\rho_a = \sum_b m_b W_{ab} \qquad (6)$$

or the continuity density given by

$$\frac{d\rho_a}{dt} = -\sum_b m_b(\bar{v}_a - \bar{v}_b)\nabla_a W_{ab} \qquad (7)$$

According to Liu & Liu (2003) both forms have their advantages and disadvantages, e.g. the summation density conserves mass exactly but the continuity density does not. However, the density summation approach needs more computational effort because it has to be evaluated before the other parameters but the continuity density can be evaluated with other physical parameters, thus saving computational time (Liu & Liu, 2003).

The internal energy equation or rate of change of thermal energy per unit mass is given by

$$\frac{du_a}{dt} = \frac{1}{2}\sum_b m_b\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)(\bar{v}_a - \bar{v}_b)\nabla_a W_{ab} \qquad (8)$$

which is only one of many ways the internal energy equation can be presented (Liu & Liu, 2003, Monaghan, 2005). The equations presented here are just one form of the SPH equations and by adding the viscous term to the Eulerian equations, the Navier-Stokes equations are fully represented.

Artificial Viscosity

Artificial viscosity is introduced to prevent the simulation from developing unphysical oscillations and stabilize the numerical algorithm when modeling shock phenomena (Liu & Liu, 2010, Monaghan, 2005). In shock simulation, the artificial viscosity implementation prevents unphysical penetration of approaching particles and provides dissipation, which converts kinetic energy into heat at the shock front. One form of the artificial viscosity term is given as

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha_\Pi \cdot c_{ab} \cdot \phi_{ab} + \beta_{ab} \cdot \phi_{ab}^2}{\rho_{ab}} & \bar{v}_{ab} \cdot \bar{x}_{ab} < 0 \\ 0 & \bar{v}_{ab} \cdot \bar{x}_{ab} \geq 0 \end{cases} \qquad (9)$$

where $\bar{v}_{ab}$ and $\bar{x}_{ab}$ are the velocity and position difference between particles a and b, and

$$\phi_{ab} = \frac{h_{ab} \cdot \bar{v}_{ab} \cdot \bar{x}_{ab}}{|x_{ab}|^2 + \varphi^2} \qquad (10)$$

are average density, sound velocity and smoothing length between particles, where ρ_{ab}, c_{ab}, h_{ab}. The parameters α_Π and β_Π are constants, typically set around 1.0 and the factor φ is set as $0.1h_{ab}$ to prevent numerical divergence.

The viscous term is finally added to the pressure terms in equation 5 to give

$$\frac{d\bar{v}_a}{dt} = -\sum_b m_b \cdot \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a W_{ab} \qquad (11)$$

Boundary Treatment

According to Liu & Liu (2003) the full exploitation of the SPH method has been progressing slowly because of problems of deficiency near or on the boundary of the computational domain. The main reason is that the integrals involved are in theory truncated by the boundary. On the boundary, field variables like pressure are directly calculated and therefore the boundary treatment in the SPH method is more challenging, but also more diversified, when compared to classical methods. Lot of papers have been published addressing this problem (Liu & Liu, 2003, Monaghan, 2005). The boundary can be represented by fixed virtual particles that either exert prescribed force to real particles inside the boundary or satisfy the same equations as the real particles and thus take part in the integration process. The exerted force can be implemented by using a Lennord-Jones force acting between the centers of the boundary particles and the real particles (Monaghan, 1994, Liu & Liu, 2003). The force is given by:

$$PB_{ij} = \begin{cases} D\left[\left(\frac{r_0}{r_{ij}} \right)^{n_1} - \left(\frac{r_0}{r_{ij}} \right)^{n_2} \right] \frac{\bar{x}_{ij}}{r_{ij}^2} & \left(\frac{r_0}{r_{ij}} \right) \le 1 \\ 0 & \left(\frac{r_0}{r_{ij}} \right) > 1 \end{cases} \qquad (12)$$

where the r_{ij} is the distance between particles, r_0 is the cutoff distance, usually selected close to initial particle spacing, n_1 and n_2 are constants that have to satisfy the condition $n_1 > n_2$ and D is chosen close to the square of the larges velocity (Monaghan, 1994, Liu & Liu, 2010). Another way to represent the boundary is to use ghost particles. In this case, particles that are close to the boundary are reflected outside the boundary with the same field properties as the real particles, but with velocity mirrored with respect to the boundary surface (Liu & Liu, 2010). The space outside the boundary is then filled with ghost particles and enables smoothed interpolation of the particles near the boundary. In the work of Liu & Liu (2010) the fixed particles that exert force are used in conjunction with the ghost boundary particles. This kind of boundary treatment has shown good results in numerical test (Liu & Liu 2010).

SPH CODE FORMULATION

The current modeling work is based on recent publications in the field and programming is done in C++. Source code for the SPH method from Liu & Liu (2003) was used as a reference when setting up the code structure and subroutines.

The Code Structure

The code structure is schematically shown in figure 1. In the main function, SPH main, an object is created, "SPH problem", which contains all problem dependent information and particle information. In the "SPH problem" object each particle is stored with it's own properties, such as position, pressure, density, etc.

The problem object, "SPH problem", is then initialized in the input module, which sets up the geometry, boundaries and dimensions, and then this object is passed to the time integration module. The time integration module then starts the simulation and for each time step calls the single step module. In the single step module the special features of the SPH method are involved. This process starts by managing the boundary particles. Then particle interaction is calculated with a nearest neighbor particle search (NNPS) algorithm, which is to be used by the smoothing function kernel. The density is approximated from either summation density or continuity density. An internal force module calculates the interaction forces on the right hand side of the Navier-Stokes equations and uses an equation of state (EOS) to relate pressure, density and internal energy. The external forces are then calculated e.g. the gravity, boundary force etc. Now, the modules are executed serially one after another and finally the momentum, energy and density values are updated for each particle. The output module saves the results for a given interval to external files at the end of each time step, that is the position, velocity, pressure etc. for all particles.

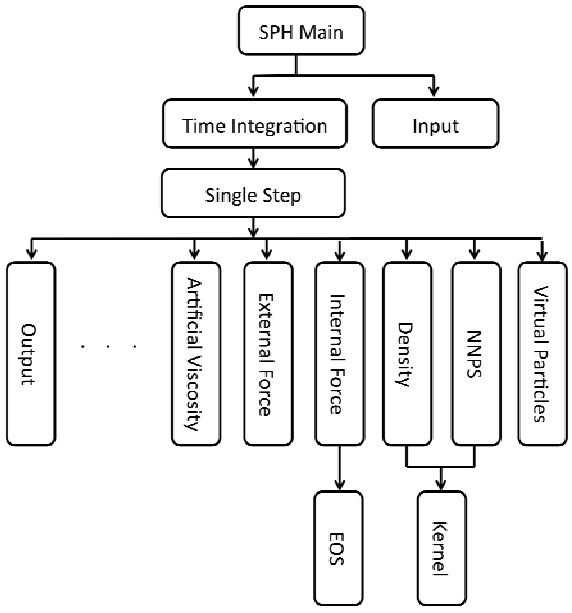

Figure 1: The structure of the SPH code

Time Integration

The numerical scheme used for time integration is a second order Leap-Frog scheme, because of it's low memory storage in the implementation and the

efficiency for one force evaluation per step (Liu & Liu, 2003). The scheme is then an explicit one, but with reasonable high accuracy. Great care must be taken in selecting the size of each timestep, since large timesteps can result in instabilities (well known problem for explicit methods) and in particular, problems on the boundary can be expected if the timestep is too large. This would bee seen as particle leakage through boundaries, which is a very serious defect in implementing the method.

Computational efficiency

In the grid based numerical methods, the neighboring grid cells are well defined but the SPH method requires that all particles within specific distance from any given particle have to be identified. These particles are called nearest neighboring particles and the process of finding them is commonly referred to as *nearest neighboring particle search* (NNPS). Instead of using direct search, where the distance between all particles is calculated, a linked-list search algorithm is applied. The linked-list search increases the computational efficiency by dividing the computational domain into boxes as wide as the smoothing length (or the support domain of each particle). Then for each particle, it is only necessary to calculate the distance within the particles own box and the nearest neighbor boxes. For N particles, the direct search algorithm needs about N^2 operations per timestep while the linked-list search algorithm needs N logN operations, thus increasing the computational efficiency (Liu & Liu, 2003). The linked-list search is though restricted to constant smoothing length in its simplest form, but can also be applied for implementations with varying smoothing length.

RESULTS

The shock tube problem is a good benchmark for numerical methods and has been used to study the SPH method (Liu & Liu, 2003, Monaghan, 2005). The shock tube is a long straight tube filled with gas with a membrane in the middle, separating the tube into two sections with different density and pressure. Both sections are at equilibrium at t=0 when the membrane breaks. Then a shockwave travels into the low-density region, followed by a contact discontinuity in the same direction and a rarefaction wave moves in the opposite direction into the high-density region. The initial conditions for this example are defined so that in the left side of the tube (x≤0), the high-density section, density, pressure and internal energy are set to 1.0, 1.0 and 2.5, respectively. For the right hand side these variables are initialized to 0.25, 0.1795, 1.795, respectively. The initial velocities are zero for both high-density and low-density sections (Liu & Liu, 2003). The Equation of state used in the simulation is the one for ideal gas, given by $p = (\gamma - 1)\rho e$ where γ=1.4.

Results For Shock Tube In One Dimension

The problem is set up in similar way as in Liu & Liu (2003) where in the simulation a total of 400 particles are used, with equal mass, m_i=0.001875. The length of the pipe is 1.2 m.

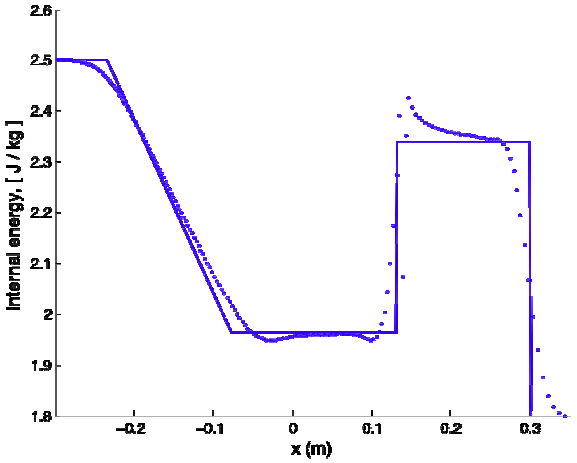

Figure 2: The internal energy, e, vs. the position. The dots are the SPH solution and the solid line the exact solution

There are 80 particles in the low-density section (from 0 to 0.6 m) equally spaced with Δx = 0.0075 and in the high-density section there are 320 particles (from -0.6 to 0 m), also equally spaced with Δx = 0.001875. The simulation ran for 0.2 s with time step of dt = 0.005 s. There are no effective boundary conditions since the area of interest is not affected by the boundary within this timeframe. In figure 2 and figure 3 the internal energy and velocities are presented, respectively.

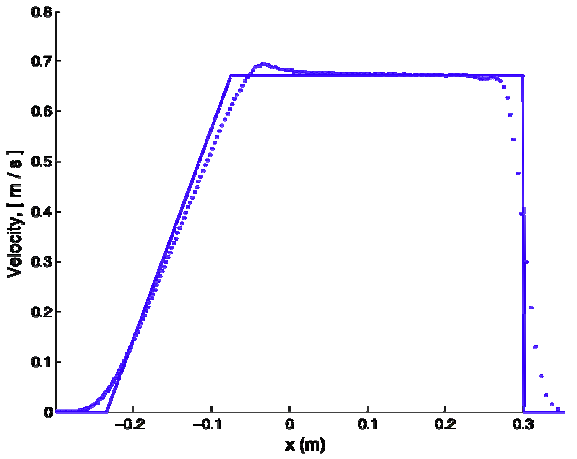

Figure 3: The velocity in the shock tube vs. position. The dots are the SPH solution and the solid line the exact solution.

The solid lines represent the exact solution and the dots are results from the SPH simulation. The pressure and velocity are constant between the rarefaction wave and the shock but the at the contact

discontinuity the density and internal energy change discontinuously (Liu & Liu, 2010, Monaghan, 2005). The methods works reasonably well, but slight overshoots are observed when changes in variables are sharp, which would be expected for any numerical method.

Result for Shock Tube In Three Dimensions

The preliminary results for the three-dimensional shock tube model are given here. The initial conditions are the same as for the one-dimensional case, except for the introduction of boundary particles, which are arranged in a tube form that is open on both ends. The dimensions are the same, except the tube has now a diameter of D=0.01. The initial arrangement of the particles is according to a face center cubic structure (FCC). As for the one-dimensional case, all particles have equal mass, m_i=2.8963e-9. The spacing between the particles in the high-pressure section is r=0.016 and for the low-pressure section, r≈0.00254. The boundary particles are of a dynamic type, that is they take part in the summation but remain fixed. The simulation ran for 0.2 s with time step of dt=0.0005s with a total of 77013 particles. In figure 4 and 5, preliminary results are presented where the particles internal energy and velocities are plotted as a function of their z position, which is in the direction of the pipe. The upper plots of figure 4 and 5 show all the gas particles in the pipe but the lower plot shows just the particles in the center of the pipe that are inside a imaginary pipe having half the diameter of the actual pipe simulated, that is 1/2 D.

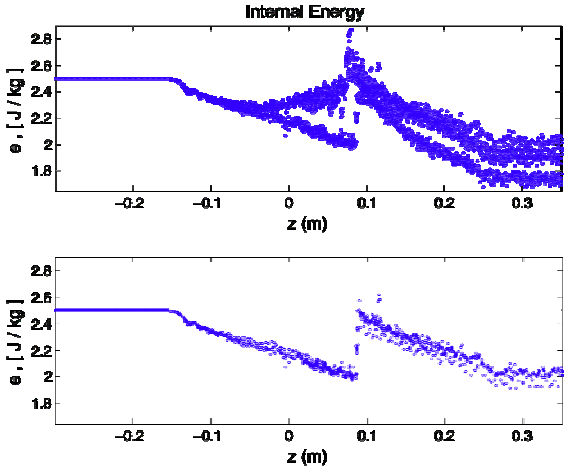

Figure 4: The internal energy, e, vs. the position. The dots show the SPH solution. In the upper plot all particles are plotted, but in the lower one, only particles at the center of the pipe are shown

If the results are compared to the results from the one-dimensional shock tube there are some resemblences, but the three-dimensional shock tube with boundary particles does not quite capture the shock physics. The shock in figure 4 and 5 is smoothed out and the velocity jump isn't high enough, but the contact discontinuity is more visible in figure 4. A possible solution for better results is to reconsider the boundary conditions as they play an important role in the three-dimensional model implementation.

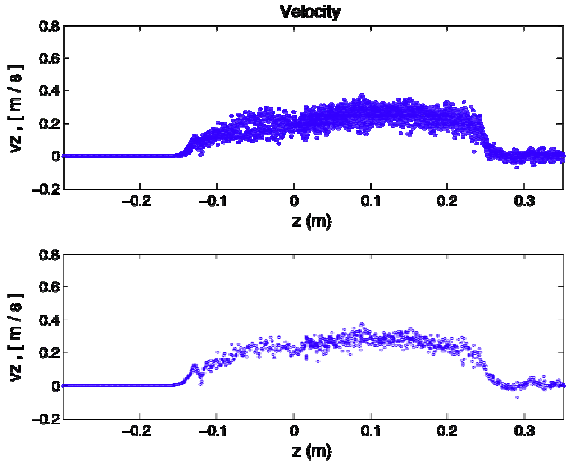

Figure 5: The internal energy, e, vs. the position. The dots show the SPH solution. In the upper plot all particles are plotted, but in the lower one only particles that are positioned in the center of the pipe are shown

FUTURE STUDIES OF TWO-PHASE FLOW

When modeling two-phase flow in a pipe with the SPH method, one needs to represent each fluid with it's own set of particles followed by a proper equation of state. For traditional grid-based methods, a slightly compressible liquid such as water can be approximated by an artificial incompressible fluid. For the SPH method, it is better to approximate it as an artificial fluid, which is moderately compressible. The most popular equation of state for water in SPH simulations is one introduced by R. H. Cole, given by (Monaghan, 2005)

$$P = B \cdot \left(\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right) \qquad (13)$$

where ρ_0 is a reference density, γ~7 and B is chosen so that the artificial speed of sound sufficiently large to make density fluctuations negligible (Monaghan, 2005, Colagrossi, 2003).

In two-phase modeling the treatment of the interface is important, as the computational information is transferred through it (Liu & Liu, 2003). Without this consideration, unphysical penetration of particles with different phase takes place through the interface boundary, resulting in excess mixing. To compensate for this, one can apply a penalty force similar to the

boundary reflection force, given in equation 12 (Liu & Liu, 2003).

CONCLUSION

The SPH method has been studied in this work and the programming framework been formulated in C++. The SPH program has been tested with the well known one-dimensional shock tube case, with good results. The problem has been extended to three-dimensions and preliminary results indicate that improvements are needed in the regions close to the boundaries (pipe walls), which is presently in progress. The next step for the work is implementing two-phase flow calculations with different particles representing each phase. The final result will be a tool that can be used to simulate two-phase flow in both pipes (wellbores or surface pipes) and steam separators.

ACKNOWLEDGEMENT

The authors wish to thank Rannsóknarsjóður Háskóla Íslands, the Geothermal Research Group (GEORG) and Orkurannsóknasjóður Landsvirkjunnar for their financial support of this project.

REFERENCES

Ballzus, C. , Karlsson, T. and Maack, R. (1992) "Design of Geothermal Steam Supply Systems in Iceland", *Geothermics*, **21**, 835-845.

Colagrossi, A. and Landrini, M. (2003), "Numerical simulation of interfacial flows by smoothed particle hydrodynamics", *Journal of Computational Physics*, **191**, 448-475.

Gingold, R.A. and Monaghan, J.J. (1977), "Smoothed particle hydrodynamics: theory and application to non-spherical stars",☐☐ *Monthly Notice Royal Astronomical Society*, **181**, 375-389.

Liu, G.R. and Liu, M.B. (2003), "Smoothed Particle Hydrodynamics", *World Scientific Pub Co Inc*.

Liu, M.B. and Liu, G.R. (2010), "Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments",☐☐ *Archives of Computational Methods in Engineering*, **17**, 25-76.

Lucy, L.B. (1977), "A numerical approach to testing of the fission hypothesis", ", *The Astronomical Journal*, **82**, 1013-1024.

Monaghan, J.J. (1992), "Smoothed Particle Hydrodynamics",☐☐ *Annual Rev. Astron. Astrophys.*, **30**, 543-574.

Monaghan, J.J. (1994), "Simulating Free Surface Flows with SPH",☐☐ *Journal of Computational Physics*, **110**, 399-406.

Monaghan, J.J. (2005), "Smoothed particle hydrodynamics", *Reports on Progress in Physics*, **68**, 1703-1759.