



WIKIPEDIA
The Free Encyclopedia

Smoothed-particle hydrodynamics

Smoothed-particle hydrodynamics (**SPH**) is a computational method used for simulating the mechanics of continuum media, such as solid mechanics and fluid flows. It was developed by Gingold and Monaghan^[2] and Lucy^[3] in 1977, initially for astrophysical problems. It has been used in many fields of research, including astrophysics, ballistics, volcanology, and oceanography. It is a meshfree Lagrangian method (where the co-ordinates move with the fluid), and the resolution of the method can easily be adjusted with respect to variables such as density.

Method

Advantages

- By construction, SPH is a meshfree method, which makes it ideally suited to simulate problems dominated by complex boundary dynamics, like free surface flows, or large boundary displacement.
- The lack of a mesh significantly simplifies the model implementation and its parallelization, even for many-core architectures.^{[4][5]}
- SPH can be easily extended to a wide variety of fields, and hybridized with some other models, as discussed in Modelling Physics.
- As discussed in section on weakly compressible SPH, the method has great conservation features.
- The computational cost of SPH simulations per number of particles is significantly less than the cost of grid-based simulations per number of cells when the metric of interest is related to fluid density (e.g., the probability density function of density fluctuations).^[6] This is the case because in SPH the resolution is put where the matter is.

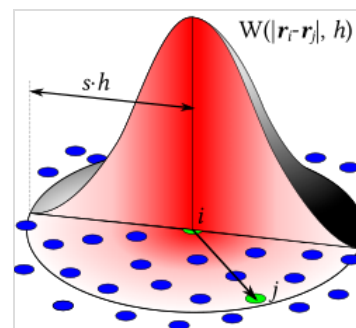
Limitations

- Setting boundary conditions in SPH such as inlets and outlets^[7] and walls^[8] is more difficult than with grid-based methods. In fact, it has been stated that "the treatment of boundary conditions is certainly one of the most difficult technical points of the SPH method".^[9] This challenge is partly because in SPH the particles near the boundary change with time.^[10] Nonetheless, wall boundary conditions for SPH are available.^{[8][10][11]}
- The computational cost of SPH simulations per number of particles is significantly larger than the cost of grid-based simulations per number of cells when the metric of interest is not (directly) related to density (e.g., the kinetic-energy spectrum).^[6] Therefore, overlooking issues of parallel speedup, the simulation of constant-density flows (e.g., external aerodynamics) is more efficient with grid-based methods than with SPH.

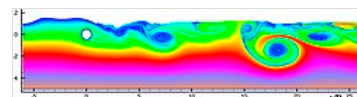
Examples

Fluid dynamics

Smoothed-particle hydrodynamics is being increasingly used to model fluid motion as well. This is due to several benefits over traditional grid-based techniques. First, SPH guarantees conservation of mass without extra computation since the particles themselves represent mass. Second, SPH computes pressure from weighted contributions of neighboring particles rather than by solving linear systems of equations. Finally, unlike grid-based techniques, which must track fluid boundaries, SPH creates a free surface for two-phase interacting fluids directly since the particles represent the denser fluid (usually water) and empty space represents the lighter fluid (usually air). For these reasons, it is possible to simulate fluid motion using SPH in real time. However, both grid-based and SPH techniques still require the generation of renderable free surface geometry using a polygonization technique such as metaballs and marching cubes, point splatting, or 'carpet' visualization. For gas dynamics it is more appropriate to use the kernel function itself to produce a rendering of gas column density (e.g., as done in the SPLASH visualisation package).



Schematic view of an SPH convolution



Flow around cylinder with free surface modelled with SPH. See^[1] for similar simulations.

One drawback over grid-based techniques is the need for large numbers of particles to produce simulations of equivalent resolution. In the typical implementation of both uniform grids and SPH particle techniques, many voxels or particles will be used to fill water volumes that are never rendered. However, accuracy can be significantly higher with sophisticated grid-based techniques, especially those coupled with particle methods (such as particle level sets), since it is easier to enforce the incompressibility condition in these systems. SPH for fluid simulation is being used increasingly in real-time animation and games where accuracy is not as critical as interactivity.

Recent work in SPH for fluid simulation has increased performance, accuracy, and areas of application:

- B. Solenthaler, 2009, develops Predictive-Corrective SPH (PCISPH) to allow for better incompressibility constraints^[12]
- M. Ihmsen et al., 2010, introduce boundary handling and adaptive time-stepping for PCISPH for accurate rigid body interactions^[13]
- K. Bodin et al., 2011, replace the standard equation of state pressure with a density constraint and apply a variational time integrator^[14]
- R. Hoetzlein, 2012, develops efficient GPU-based SPH for large scenes in Fluids v.3^[15]
- N. Akinci et al., 2012, introduce a versatile boundary handling and two-way SPH-rigid coupling technique that is completely based on hydrodynamic forces; the approach is applicable to different types of SPH solvers^[16]
- M. Macklin et al., 2013 simulates incompressible flows inside the Position Based Dynamics framework, for bigger timesteps^[17]
- N. Akinci et al., 2013, introduce a versatile surface tension and two-way fluid-solid adhesion technique that allows simulating a variety of interesting physical effects that are observed in reality^[18]
- J. Kyle and E. Terrell, 2013, apply SPH to Full-Film Lubrication^[19]
- A. Mahdavi and N. Talebbeydokhti, 2015, propose a hybrid algorithm for implementation of solid boundary condition and simulate flow over a sharp crested weir^[20]
- S. Tavakkol et al., 2016, develop curvSPH, which makes the horizontal and vertical size of particles independent and generates uniform mass distribution along curved boundaries^[21]
- W. Kistorz and A. Esmail-Yakas, 2020, propose a general, efficient and simple method for evaluating normalization factors near piecewise-planar boundaries^[11]
- Colagrossi et al., 2019, study flow around a cylinder close to a free-surface and compare with other techniques^[1]

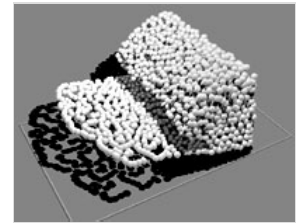


Fig. SPH simulation of ocean waves using FLUIDS v.1 (Hoetzlein)

Astrophysics

Smoothed-particle hydrodynamics's adaptive resolution, numerical conservation of physically conserved quantities, and ability to simulate phenomena covering many orders of magnitude make it ideal for computations in theoretical astrophysics.^[22]

Simulations of galaxy formation, star formation, stellar collisions,^[23] supernovae^[24] and meteor impacts are some of the wide variety of astrophysical and cosmological uses of this method.

SPH is used to model hydrodynamic flows, including possible effects of gravity. Incorporating other astrophysical processes which may be important, such as radiative transfer and magnetic fields is an active area of research in the astronomical community, and has had some limited success.^{[25][26]}

Solid mechanics

Libersky and Petschek^{[27][28]} extended SPH to Solid Mechanics. The main advantage of SPH in this application is the possibility of dealing with larger local distortion than grid-based methods. This feature has been exploited in many applications in Solid Mechanics: metal forming, impact, crack growth, fracture, fragmentation, etc.

Another important advantage of meshfree methods in general, and of SPH in particular, is that mesh dependence problems are naturally avoided given the meshfree nature of the method. In particular, mesh alignment is related to problems involving cracks and it is avoided in SPH due to the isotropic support of the kernel functions. However, classical SPH formulations suffer from tensile instabilities^[29] and lack of consistency.^[30] Over the past years, different corrections have been introduced to improve the accuracy of the SPH solution, leading to the RKPM by Liu et al.^[31] Randles and Libersky^[32] and Johnson and Beissel^[33] tried to solve the consistency problem in their study of impact phenomena.

Dyka et al.^{[34][35]} and Randles and Libersky^[36] introduced the stress-point integration into SPH and Ted Belytschko et al.^[37] showed that the stress-point technique removes the instability due to spurious singular modes, while tensile instabilities can be avoided by using a Lagrangian kernel. Many other recent studies can be found in the literature devoted to improve the convergence of the SPH method.

Recent improvements in understanding the convergence and stability of SPH have allowed for more widespread applications in Solid Mechanics. Other examples of applications and developments of the method include:

- Metal forming simulations.^[38]
- SPH-based method SPAM (Smoothed Particle Applied Mechanics) for impact fracture in solids by [William G. Hoover](#).^[39]
- Modified SPH (SPH/MLSPH) for fracture and fragmentation.^[40]
- Taylor-SPH (TSPH) for shock wave propagation in solids.^[41]
- Generalized coordinate SPH (GSPH) allocates particles inhomogeneously in the Cartesian coordinate system and arranges them via mapping in a generalized coordinate system in which the particles are aligned at a uniform spacing.^[42]

Numerical tools

Interpolations

The Smoothed-Particle Hydrodynamics (SPH) method works by dividing the fluid into a set of discrete moving elements \mathbf{i}, \mathbf{j} , referred to as particles. Their Lagrangian nature allows setting their position \mathbf{r}_i by integration of their velocity \mathbf{v}_i as:

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i.$$

These particles interact through a [kernel function](#) with characteristic radius known as the "smoothing length", typically represented in equations by h . This means that the physical quantity of any particle can be obtained by summing the relevant properties of all the particles that lie within the range of the kernel, the latter being used as a weighting function W . This can be understood in two steps. First an arbitrary field A is written as a convolution with W :

$$A(\mathbf{r}) = \int A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h) dV(\mathbf{r}').$$

The error in making the above approximation is order h^2 . Secondly, the integral is approximated using a Riemann summation over the particles:

$$A(\mathbf{r}) = \sum_j V_j A_j W(|\mathbf{r} - \mathbf{r}_j|, h),$$

where the summation over j includes all particles in the simulation. V_j is the [volume](#) of particle j , A_j is the value of the quantity A for particle j and \mathbf{r} denotes position. For example, the density ρ_i of particle i can be expressed as:

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W_{ij},$$

where $m_j = \rho_j V_j$ denotes the particle mass and ρ_j the particle density, while $W_{ij} = W_{ji}$ is a short notation for $W(|\mathbf{r}_i - \mathbf{r}_j|, h)$. The error done in approximating the integral by a discrete sum depends on h , on the particle size (i.e. $V_j^{1/d}$, d being the space dimension), and on the particle arrangement in space. The latter effect is still poorly known.^[43]

Kernel functions commonly used include the [Gaussian function](#), the [quintic spline](#) and the Wendland C^2 kernel.^[44] The latter two kernels are compactly supported (unlike the Gaussian, where there is a small contribution at any finite distance away), with support proportional to h . This has the advantage of saving computational effort by not including the relatively minor contributions from distant particles.

Although the size of the smoothing length can be fixed in both [space](#) and [time](#), this does not take advantage of the full power of SPH. By assigning each particle its own smoothing length and allowing it to vary with time, the resolution of a simulation can be made to automatically adapt itself depending on local conditions. For example, in a very dense region where many particles are close together, the smoothing length can be made relatively short, yielding high spatial resolution. Conversely, in low-density regions where individual particles are far apart and the resolution is low, the smoothing length can be increased, optimising the computation for the regions of interest.

Discretization of governing equations

For particles of constant mass, differentiating the interpolated density ρ_i with respect to time yields

$$\frac{d\rho_i}{dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij},$$

where $\nabla W_{ij} = -\nabla W_{ji}$ is the gradient of W_{ij} with respect to \mathbf{r}_i . Comparing this equation with the continuity equation in the Lagrangian description (using material derivatives),

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

it is apparent that its right-hand side is an approximation of $-\rho \nabla \cdot \mathbf{v}$; hence one defines a discrete divergence operator as follows:

$$\mathbf{D}_i \{\mathbf{v}_j\} = -\frac{1}{\rho_i} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij}.$$

This operator gives an SPH approximation of $\nabla \cdot \mathbf{v}$ at the particle i for a given set of particles with given masses m_j , positions $\{\mathbf{r}_j\}$ and velocities $\{\mathbf{v}_j\}$.

The other important equation for a compressible inviscid fluid is the Euler equation for momentum balance:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \mathbf{g}$$

Similarly to continuity, the task is to define a discrete gradient operator in order to write

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho} \mathbf{G}_i \{p_j\} + \mathbf{g}$$

One choice is

$$\mathbf{G}_i \{p_j\} = \rho_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij},$$

which has the property of being skew-adjoint with the divergence operator above, in the sense that

$$\sum_i V_i \mathbf{v}_i \cdot \mathbf{G}_i \{p_j\} = -\sum_i V_i p_i \mathbf{D}_i \{\mathbf{v}_j\},$$

this being a discrete version of the continuum identity

$$\int \mathbf{v} \cdot \mathbf{grad} p = -\int p \mathbf{div} \cdot \mathbf{v}.$$

This property leads to nice conservation properties.^[45]

Notice also that this choice leads to a symmetric divergence operator and antisymmetric gradient. Although there are several ways of discretizing the pressure gradient in the Euler equations, the above antisymmetric form is the most acknowledged one. It supports strict conservation of linear and angular momentum. This means that a force that is exerted on particle i by particle j equals the one that is exerted on particle j by particle i including the sign change of the effective direction, thanks to the antisymmetry property $\nabla W_{ij} = -\nabla W_{ji}$.

Nevertheless, other operators have been proposed, which may perform better numerically or physically. For instance, one drawback of these operators is that while the divergence \mathbf{D} is zero-order consistent (i.e. yields zero when applied to a constant vector field), it can be seen that the gradient \mathbf{G} is not. Several techniques have been proposed to circumvent this issue, leading to renormalized operators (see e.g.^[46]).

Variational principle

The above SPH governing equations can be derived from a least action principle, starting from the Lagrangian of a particle system:

$$\mathcal{L} = \sum_j m_j \left(\frac{1}{2} \mathbf{v}_j^2 - e_j + \mathbf{g} \cdot \mathbf{r}_j \right),$$

where e_j is the particle specific internal energy. The Euler–Lagrange equation of variational mechanics reads, for each particle:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \mathbf{v}_i} = \frac{\partial \mathcal{L}}{\partial \mathbf{r}_i}.$$

When applied to the above Lagrangian, it gives the following momentum equation:

$$m_i \frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \frac{\partial e_j}{\partial \mathbf{r}_i} + m_i \mathbf{g} = - \sum_j m_j \frac{\partial e_j}{\partial \rho_j} \frac{\partial \rho_j}{\partial \mathbf{r}_i} + m_i \mathbf{g}$$

where the chain rule has been used, since e_j depends on ρ_j , and the latter, on the position of the particles. Using the thermodynamic property $d\mathbf{e} = (p/\rho^2) d\rho$ we may write

$$m_i \frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \frac{p_j}{\rho_j^2} \frac{\partial \rho_j}{\partial \mathbf{r}_i} + m_i \mathbf{g},$$

Plugging the SPH density interpolation and differentiating explicitly $\frac{\partial \rho_j}{\partial \mathbf{r}_i}$ leads to

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} + \mathbf{g},$$

which is the SPH momentum equation already mentioned, where we recognize the \mathbf{G} operator. This explains why linear momentum is conserved, and allows conservation of angular momentum and energy to be conserved as well.^[47]

Time integration

From the work done in the 80's and 90's on numerical integration of point-like particles in large accelerators, appropriate time integrators have been developed with accurate conservation properties on the long term; they are called symplectic integrators. The most popular in the SPH literature is the leapfrog scheme, which reads for each particle i :

$$\begin{aligned} \mathbf{v}_i^{n+1/2} &= \mathbf{v}_i^n + \mathbf{a}_i^n \frac{\Delta t}{2}, \\ \mathbf{r}_i^{n+1} &= \mathbf{r}_i^n + \mathbf{v}_i^{i+1/2} \Delta t, \\ \mathbf{v}_i^{n+1} &= \mathbf{v}_i^{n+1/2} + \mathbf{a}_i^{i+1} \frac{\Delta t}{2}, \end{aligned}$$

where Δt is the time step, superscripts stand for time iterations while \mathbf{a}_i is the particle acceleration, given by the right-hand side of the momentum equation.

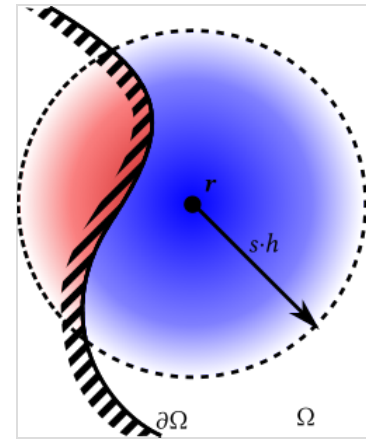
Other symplectic integrators exist (see the reference textbook^[48]). It is recommended to use a symplectic (even low-order) scheme instead of a high order non-symplectic scheme, to avoid error accumulation after many iterations.

Integration of density has not been studied extensively (see below for more details).

Symplectic schemes are conservative but explicit, thus their numerical stability requires stability conditions, analogous to the Courant-Friedrichs-Lewy condition (see below).

Boundary techniques

In case the SPH convolution shall be practiced close to a boundary, i.e. closer than $s \cdot h$, then the integral support is truncated. Indeed, when the convolution is affected by a boundary, the convolution shall be split in 2 integrals,



SPH Convolution support split close to a boundary

$$A(\mathbf{r}) = \int_{\Omega(\mathbf{r})} A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h) d\mathbf{r}' + \int_{B(\mathbf{r}) - \Omega(\mathbf{r})} A(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h) d\mathbf{r}',$$

where $B(\mathbf{r})$ is the compact support ball centered at \mathbf{r} , with radius $s \cdot h$, and $\Omega(\mathbf{r})$ denotes the part of the compact support inside the computational domain, $\Omega \cap B(\mathbf{r})$. Hence, imposing boundary conditions in SPH is completely based on approximating the second integral on the right hand side. The same can be of course applied to the differential operators computation,

$$\nabla A(\mathbf{r}) = \int_{\Omega(\mathbf{r})} A(\mathbf{r}') \nabla W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' + \int_{B(\mathbf{r}) - \Omega(\mathbf{r})} A(\mathbf{r}') \nabla W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}'.$$

Several techniques has been introduced in the past to model boundaries in SPH.

Integral neglect

The most straightforward boundary model is neglecting the integral,

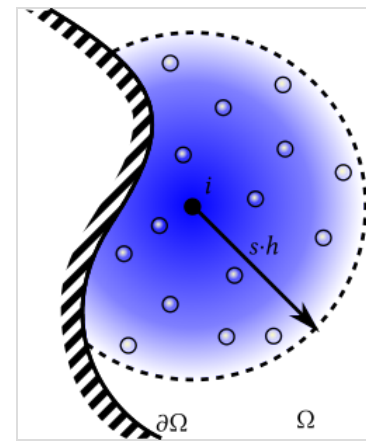
$$\int_{B(\mathbf{r}) - \Omega(\mathbf{r})} A(\mathbf{r}') \nabla W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' \simeq \mathbf{0},$$

such that just the bulk interactions are taken into account,

$$\nabla A_i = \sum_{j \in \Omega_i} V_j A_j \nabla W_{ij}.$$

This is a popular approach when free-surface is considered in monophase simulations.^[49]

The main benefit of this boundary condition is its obvious simplicity. However, several consistency issues shall be considered when this boundary technique is applied.^[49] That's in fact a heavy limitation on its potential applications.



SPH free-surface model by means of integral neglect

Fluid Extension

Probably the most popular methodology, or at least the most traditional one, to impose boundary conditions in SPH, is Fluid Extension technique. Such technique is based on populating the compact support across the boundary with so-called ghost particles, conveniently imposing their field values.^[50]

Along this line, the integral neglect methodology can be considered as a particular case of fluid extensions, where the field, A, vanish outside the computational domain.

The main benefit of this methodology is the simplicity, provided that the boundary contribution is computed as part of the bulk interactions. Also, this methodology has been deeply analyzed in the literature.^{[51][50][52]}

On the other hand, deploying ghost particles in the truncated domain is not a trivial task, such that modelling complex boundary shapes

becomes cumbersome. The 2 most popular approaches to populate the empty domain with ghost particles are Mirrored-Particles^[53] and Fixed-Particles.^[50]

Boundary Integral

The newest Boundary technique is the Boundary Integral methodology.^[54] In this methodology, the empty volume integral is replaced by a surface integral, and a renormalization:

$$\nabla A_i = \frac{1}{\gamma_i} \left(\sum_{j \in \Omega_i} V_j A_j \nabla W_{ij} + \sum_{j \in \partial \Omega_i} S_j A_j \mathbf{n}_j W_{ij} \right),$$

$$\gamma_i = \sum_{j \in \Omega_i} V_j W_{ij},$$

with \mathbf{n}_j the normal of the generic j -th boundary element. The surface term can be also solved considering a semi-analytic expression.^[54]

Modelling physics

Hydrodynamics

Weakly compressible approach

Another way to determine the density is based on the SPH smoothing operator itself. Therefore, the density is estimated from the particle distribution utilizing the SPH interpolation. To overcome undesired errors at the free surface through kernel truncation, the density formulation can again be integrated in time.^[54]

The weakly compressible SPH in fluid dynamics is based on the discretization of the Navier–Stokes equations or Euler equations for compressible fluids. To close the system, an appropriate equation of state is utilized to link pressure \mathbf{p} and density ρ . Generally, the so-called Cole equation^[55] (sometimes mistakenly referred to as the "Tait equation") is used in SPH. It reads

$$\mathbf{p} = \frac{\rho_0 c^2}{\gamma} \left(\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right) + \mathbf{p}_0,$$

where ρ_0 is the reference density and c the speed of sound. For water, $\gamma = 7$ is commonly used. The background pressure \mathbf{p}_0 is added to avoid negative pressure values.

Real nearly incompressible fluids such as water are characterized by very high speeds of sound of the order 10^3 m/s . Hence, pressure information travels fast compared to the actual bulk flow, which leads to very small Mach numbers M . The momentum equation leads to the following relation:

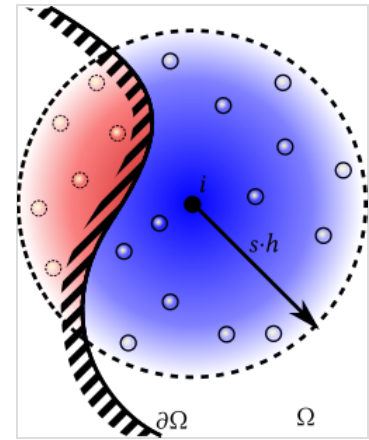
$$\frac{\delta \rho}{\rho_0} \approx \frac{|\mathbf{v}|^2}{c^2} = M^2$$

where ρ is the density change and \mathbf{v} the velocity vector. In practice a value of c smaller than the real one is adopted to avoid time steps too small in the time integration scheme. Generally a numerical speed of sound is adopted such that density variation smaller than 1% are allowed. This is the so-called weak-compressibility assumption. This corresponds to a Mach number smaller than 0.1, which implies:

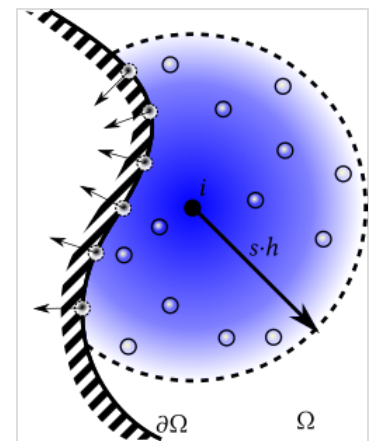
$$c = 10 v_{\max}$$

where the maximum velocity v_{\max} needs to be estimated, for e.g. by Torricelli's law or an educated guess. Since only small density variations occur, a linear equation of state can be adopted.^[56]

$$\mathbf{p} = c^2 (\rho - \rho_0)$$



SPH Fluid Extension Boundary technique



SPH Boundary Integral model

Usually the weakly-compressible schemes are affected by a high-frequency spurious noise on the pressure and density fields.^[57] This phenomenon is caused by the nonlinear interaction of acoustic waves and by fact that the scheme is explicit in time and centered in space.^[58]

Through the years, several techniques have been proposed to get rid of this problem. They can be classified in three different groups:

1. the schemes that adopt density filters,
2. the models that add a diffusive term in the continuity equation,
3. the schemes that employ Riemann solvers to model the particle interaction.

Density filter technique

The schemes of the first group apply a filter directly on the density field to remove the spurious numerical noise. The most used filters are the MLS (moving least squares) and the Shepard filter^[57] which can be applied at each time step or every n time steps. The more frequent is the use of the filtering procedure, the more regular density and pressure fields are obtained. On the other hand, this leads to an increase of the computational costs. In long time simulations, the use of the filtering procedure may lead to the disruption of the hydrostatic pressure component and to an inconsistency between the global volume of fluid and the density field. Further, it does not ensure the enforcement of the dynamic free-surface boundary condition.

Diffusive term technique

A different way to smooth out the density and pressure field is to add a diffusive term inside the continuity equation (group 2) :

$$\frac{d\rho_i}{dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij} + \mathcal{D}_i(\rho),$$

The first schemes that adopted such an approach were described in Ferrari^[59] and in Molteni^[56] where the diffusive term was modeled as a Laplacian of the density field. A similar approach was also used in Fatehi and Manzari.^[60]

In Antuono et al.^[61] a correction to the diffusive term of Molteni^[56] was proposed to remove some inconsistencies close to the free-surface. In this case the adopted diffusive term is equivalent to a high-order differential operator on the density field.^[62] The scheme is called δ -SPH and preserves all the conservation properties of the SPH without diffusion (e.g., linear and angular momenta, total energy, see^[63]) along with a smooth and regular representation of the density and pressure fields.

In the third group there are those SPH schemes which employ numerical fluxes obtained through Riemann solvers to model the particle interactions.^{[64][65][66]}

Riemann solver technique

For an SPH method based on Riemann solvers, an inter-particle Riemann problem is constructed along a unit vector $\mathbf{e}_{ij} = -\mathbf{r}_{ij}/r_{ij}$ pointing from particle i to particle j . In this Riemann problem the initial left and right states are on particles i and j , respectively. The \mathbf{L} and \mathbf{R} states are

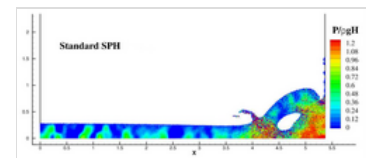
$$\begin{cases} (\rho_L, U_L, P_L) = (\rho_i, \mathbf{v}_i \cdot \mathbf{e}_{ij}, P_i) \\ (\rho_R, U_R, P_R) = (\rho_j, \mathbf{v}_j \cdot \mathbf{e}_{ij}, P_j). \end{cases}$$

The solution of the Riemann problem results in three waves emanating from the discontinuity. Two waves, which can be shock or rarefaction wave, traveling with the smallest or largest wave speed. The middle wave is always a contact discontinuity and separates two intermediate states, denoted by (ρ_L^*, U_L^*, P_L^*) and (ρ_R^*, U_R^*, P_R^*) . By assuming that the intermediate state satisfies $U_L^* = U_R^* = U^*$ and $P_L^* = P_R^* = P^*$, a linearized Riemann solver for smooth flows or with only moderately strong shocks can be written as

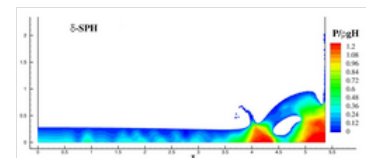
$$\begin{cases} U^* = \bar{U} + \frac{1}{2} \frac{(P_L - P_R)}{\bar{\rho} c_0} \\ P^* = \bar{P} + \frac{1}{2} \bar{\rho} c_0 (U_L - U_R), \end{cases}$$

where $\bar{U} = (U_L + U_R)/2$ and $\bar{P} = (P_L + P_R)/2$ are inter-particle averages. With the solution of the Riemann problem, i.e. U^* and P^* , the discretization of the SPH method is

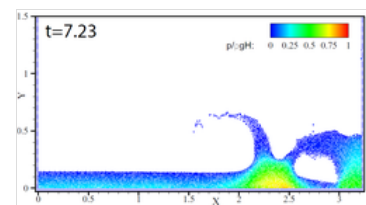
$$d\rho_i = \dots \dots m_j \dots \dots$$



SPH simulation: pressure distribution of a dam-break flow using standard SPH formulation



SPH simulation: pressure distribution of a dam-break flow using standard delta-SPH formulation



SPH simulation: pressure distribution of a dam-break flow using Riemann solver with the low-dissipation limiter.

$$\frac{d\mathbf{v}_i}{dt} = 2\rho_i \sum_j \frac{1}{\rho_j} (\mathbf{v}_i - \mathbf{v}^*) \cdot \nabla_i W_{ij},$$

$$\frac{d\mathbf{v}_i}{dt} = -2 \sum_j m_j \left(\frac{P^*}{\rho_i \rho_j} \right) \nabla_i W_{ij}.$$

where $\mathbf{v}^* = U^* \mathbf{e}_{ij} + (\bar{\mathbf{v}}_{ij} - \bar{U} \mathbf{e}_{ij})$. This indicates that the inter-particle average velocity and pressure are simply replaced by the solution of the Riemann problem. By comparing both it can be seen that the intermediate velocity and pressure from the inter-particle averages amount to implicit dissipation, i.e. density regularization and numerical viscosity, respectively.

Since the above discretization is very dissipative a straightforward modification is to apply a limiter to decrease the implicit numerical dissipations introduced by limiting the intermediate pressure by ^[67]

$$P^* = \bar{P} + \frac{1}{2} \beta \bar{\rho} (U_L - U_R),$$

where the limiter is defined as

$$\beta = \min(\eta \max(U_L - U_R, 0), \bar{c}).$$

Note that β ensures that there is no dissipation when the fluid is under the action of an expansion wave, i.e. $U_L < U_R$, and that the parameter η , is used to modulate dissipation when the fluid is under the action of a compression wave, i.e. $U_L \geq U_R$. Numerical experiments found the $\eta = 3$ is generally effective. Also note that the dissipation introduced by the intermediate velocity is not limited.

Incompressible approach

Viscosity modelling

In general, the description of hydrodynamic flows require a convenient treatment of diffusive processes to model the viscosity in the Navier–Stokes equations. It needs special consideration because it involves the Laplacian differential operator. Since the direct computation does not provide satisfactory results, several approaches to model the diffusion have been proposed.

■ Artificial viscosity

Introduced by Monaghan and Gingold ^[68] the artificial viscosity was used to deal with high Mach number fluid flows. It reads

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} \phi_{ij} + \beta \phi_{ij}^2}{\bar{\rho}_{ij}} & \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0 \\ 0 & \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} \geq 0 \end{cases}$$

Here, α is controlling a volume viscosity while β acts similar to the Neumann Richtmeyr artificial viscosity. The ϕ_{ij} is defined by

$$\phi_{ij} = \frac{h \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|^2 + \eta_h^2},$$

where η_h is a small fraction of h (e.g. $0.01h$) to prevent possible numerical infinities at close distances.

The artificial viscosity also has shown to improve the overall stability of general flow simulations. Therefore, it is applied to inviscid problems in the following form

$$\Pi_{ij} = \alpha h c \frac{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|^2 + \eta_h^2}.$$

It is possible to not only stabilize inviscid simulations but also to model the physical viscosity by this approach. To do so

$$\alpha h c = 2(n + 2) \frac{\mu}{\rho}$$

is substituted in the equation above, where n is the number of spatial dimensions of the model. This approach introduces the bulk viscosity $\zeta = \frac{5}{3} \mu$.

- Morris

For low [Reynolds numbers](#) the viscosity model by Morris ^[69] was proposed.

$$[\nu\Delta\mathbf{v}]_{ij} = 2\nu \frac{m_j}{\rho_j} \frac{\mathbf{r}_{ij} \cdot \nabla w_{h,ij}}{\|\mathbf{r}_{ij}\|^2 + \eta_h^2} \mathbf{v}_{ij}.$$

- LoShao

Additional physics

- Surface tension
- Heat transfer
- Turbulence

Multiphase extensions

Astrophysics

Often in astrophysics, one wishes to model self-gravity in addition to pure hydrodynamics. The particle-based nature of SPH makes it ideal to combine with a particle-based gravity solver, for instance [tree gravity code](#),^[70] [particle mesh](#), or [particle-particle particle-mesh](#).

Solid mechanics and fluid-structure interaction (FSI)

Total Lagrangian formulation for solid mechanics

To discretize the governing equations of solid dynamics, a correction matrix \mathbb{B}^0 ^[71] ^[72] is first introduced to reproducing rigid-body rotation as

$$\mathbb{B}_a^0 = \left(\sum_b V_b^0 (\mathbf{r}_b^0 - \mathbf{r}_a^0) \otimes \nabla_a^0 W_{ab} \right)^{-1} \quad (1)$$

where

$$\nabla_a^0 W_a = \frac{\partial W(|\mathbf{r}_{ab}^0|, h)}{\partial |\mathbf{r}_{ab}^0|} \mathbf{e}_{ab}^0$$

stands for the gradient of the kernel function evaluated at the initial reference configuration. Note that subscripts \mathbf{a} and \mathbf{b} are used to denote solid particles, and smoothing length h is identical to that in the discretization of fluid equations.

Using the initial configuration as the reference, the solid density is directly evaluated as

$$\rho_a = \rho_a^0 \frac{1}{J} \quad (2)$$

where $J = \det(\mathbb{F})$ is the Jacobian determinant of deformation tensor \mathbb{F} .

We can now discretize the momentum equation in the following form

$$m_a \frac{d\mathbf{v}}{dt} = 2 \sum_b V_a V_b \tilde{\mathbb{P}}_{ab} \nabla_a^0 W_{ab} + \mathbf{g} + \mathbf{f}_a^{F:p} + \mathbf{f}_a^{F:v} \quad (3)$$

where inter-particle averaged first Piola-Kirchhoff stress $\tilde{\mathbb{P}}$ is defined as

$$\tilde{\mathbb{P}}_{ab} = \frac{1}{2} (\mathbb{P}_a \mathbb{B}_a^0 + \mathbb{P}_b \mathbb{B}_b^0). \quad (4)$$

Also $\mathbf{f}_a^{F:p}$ and $\mathbf{f}_a^{F:v}$ correspond to the fluid pressure and viscous forces acting on the solid particle \mathbf{a} , respectively.

Fluid-structure coupling

In fluid-structure coupling, the surrounding solid structure is behaving as a moving boundary for fluid, and the no-slip boundary condition is imposed at the fluid-structure interface. The interaction forces $\mathbf{f}_i^{S:p}$ and $\mathbf{f}_i^{S:v}$ acting on a fluid particle \mathbf{i} , due to the presence of the

neighboring solid particle \mathbf{a} , can be obtained as ^[73]

$$\mathbf{f}_i^{S:p} = -2 \sum_a V_i V_a \frac{p_i \rho_a^d + p_a^d \rho_i}{\rho_i + \rho_a^d} \nabla_i W(\mathbf{r}_{ia}, h) \quad (5)$$

and

$$\mathbf{f}_i^{S:v} = 2 \sum_a \eta V_i V_a \frac{\mathbf{v}_i - \mathbf{v}_a^d}{r_{ia}} \frac{\partial W(\mathbf{r}_{ia}, h)}{\partial r_{ia}}. \quad (6)$$

Here, the imaginary pressure p_a^d and velocity \mathbf{v}_a^d are defined by

$$\begin{cases} p_a^d = p_i + \rho_i \max(0, (\mathbf{g} - \frac{d\mathbf{v}_a}{dt}) \cdot \mathbf{n}^S)(\mathbf{r}_{ia} \cdot \mathbf{n}^S) \\ \mathbf{v}_a^d = 2\mathbf{v}_i - \mathbf{v}_a. \end{cases} \quad (7)$$

where \mathbf{n}^S denotes the surface normal direction of the solid structure, and the imaginary particle density ρ_a^d is calculated through the equation of state.

Accordingly, the interaction forces $\mathbf{f}_a^{F:p}$ and $\mathbf{f}_a^{F:v}$ acting on a solid particle \mathbf{a} are given by

$$\mathbf{f}_a^{F:p} = -2 \sum_i V_a V_i \frac{p_a^d \rho_i + p_i \rho_a^d}{\rho_i + \rho_a^d} \nabla_a W(\mathbf{r}_{ai}, h) \quad (8)$$

and

$$\mathbf{f}_a^{F:v} = 2 \sum_i \eta V_a V_i \frac{\mathbf{v}_a^d - \mathbf{v}_i}{r_{ia}} \frac{\partial W(\mathbf{r}_{ia}, h)}{\partial r_{ai}}. \quad (9)$$

The anti-symmetric property of the derivative of the kernel function will ensure the momentum conservation for each pair of interacting particles \mathbf{i} and \mathbf{a} .

Others

The discrete element method, used for simulating granular materials, is related to SPH.

Variants of the method

References

- Colagrossi (2019). "Viscous flow past a cylinder close to a free surface: benchmarks with steady, periodic and metastable responses, solved by meshfree and mesh-based schemes". *Computers and Fluids*. **181**: 345–363. doi:10.1016/j.compfluid.2019.01.007 (https://doi.org/10.1016%2Fj.compfluid.2019.01.007). S2CID 128143912 (https://api.semanticscholar.org/CorpusID:128143912).
- Gingold, Robert A.; Monaghan, Joseph J. (1977). "Smoothed particle hydrodynamics: theory and application to non-spherical stars" (https://doi.org/10.1093%2Fmnras%2F181.3.375). *Monthly Notices of the Royal Astronomical Society*. **181** (3): 375–89. Bibcode:1977MNRAS.181..375G (https://ui.adsabs.harvard.edu/abs/1977MNRAS.181..375G). doi:10.1093/mnras/181.3.375 (https://doi.org/10.1093%2Fmnras%2F181.3.375).
- L.B. Lucy (1977). "A numerical approach to the testing of the fission hypothesis". *Astron. J.* **82**: 1013–1024. Bibcode:1977AJ.....82.1013L (https://ui.adsabs.harvard.edu/abs/1977AJ.....82.1013L). doi:10.1086/112164 (https://doi.org/10.1086%2F112164).
- Takahiro Harada; Seiichi Koshizuka; Yoichiro Kawaguchi (2007). *Smoothed particle hydrodynamics on GPUs*. *Computer Graphics International*. pp. 63–70.
- Alejandro Crespo; Jose M. Dominguez; Anxo Barreiro; Moncho Gomez-Gesteira; Benedict D. Rogers (2011). "GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3113801). *PLOS ONE*. **6** (6): e20685. Bibcode:2011PLoSO..620685C (https://ui.adsabs.harvard.edu/abs/2011PLoSO..620685C). doi:10.1371/journal.pone.0020685 (https://doi.org/10.1371%2Fjournal.pone.0020685). PMC 3113801 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3113801). PMID 21695185 (https://pubmed.ncbi.nlm.nih.gov/21695185).

6. Price, D. J. (2011). "Smoothed Particle Hydrodynamics: Things I wish my mother taught me". *Advances in Computational Astrophysics: Methods*. **453**: 249. arXiv:1111.1259 (https://arxiv.org/abs/1111.1259). Bibcode:2012ASPC..453..249P (https://ui.adsabs.harvard.edu/abs/2012ASPC..453..249P).
7. "The Smoothed Particle Hydrodynamics Method vs. Finite Volume Numerical Methods" (https://blog.altair.com/the-smoothed-particle-hydrodynamics-method-vs-finite-volume-numerical-methods/). 2018-03-21. Retrieved 2018-08-30.
8. Adami, S. and Hu, X. Y. and Adams, N. A.. (2012). "A generalized wall boundary condition for smoothed particle hydrodynamics". *Journal of Computational Physics*. **231** (21): 7057–7075. Bibcode:2012JCoPh.231.7057A (https://ui.adsabs.harvard.edu/abs/2012JCoPh.231.7057A). doi:10.1016/j.jcp.2012.05.005 (https://doi.org/10.1016%2Fj.jcp.2012.05.005).
9. Shadloo, M. S. and Oger, G. and Touze, D. L.. (2016). "Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges". *Computers and Fluids*. **136**: 11–34. doi:10.1016/j.compfluid.2016.05.029 (https://doi.org/10.1016%2Fj.compfluid.2016.05.029).
10. Fraser, K. and Kiss, L. I. and St-George, L. (2016). "A generalized wall boundary condition for smoothed particle hydrodynamics". *14th International LS-DYNA Conference*.
11. Kostorz (2020). "A semi-analytical boundary integral method for radial functions with application to Smoothed Particle Hydrodynamics". *Journal of Computational Physics*. **417**: 109565. Bibcode:2020JCoPh.41709565K (https://ui.adsabs.harvard.edu/abs/2020JCoPh.41709565K). doi:10.1016/j.jcp.2020.109565 (https://doi.org/10.1016%2Fj.jcp.2020.109565). S2CID 219438340 (https://api.semanticscholar.org/CorpusID:219438340).
12. Solenthaler (2009). "Predictive-Corrective Incompressible SPH". {{cite journal}}: Cite journal requires |journal= (help)
13. Imhsen (2010). "Boundary handling and adaptive time-stepping for PCISPH". *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*.
14. Bodin (2011). "Constraint Fluids" (http://www.physics.umu.se/english/research/statistical-physics-and-networks/complex-mechanical-systems/fluids-and-solids/). *IEEE Transactions on Visualization and Computer Graphics*. **18** (3): 516–26. doi:10.1109/TVCG.2011.29 (https://doi.org/10.1109%2FTVCG.2011.29). PMID 22241284 (https://pubmed.ncbi.nlm.nih.gov/22241284). S2CID 14023161 (https://api.semanticscholar.org/CorpusID:14023161).
15. Hoetzlein (2012). "Fluids v.3, A Large scale, Open Source Fluid Simulator". {{cite journal}}: Cite journal requires |journal= (help)
16. Akinci (2012). "Versatile Rigid-Fluid Coupling for Incompressible SPH". *ACM Transactions on Graphics*. **31** (4): 1–8. doi:10.1145/2185520.2185558 (https://doi.org/10.1145%2F2185520.2185558). S2CID 5669154 (https://api.semanticscholar.org/CorpusID:5669154).
17. Macklin (2013). "Position Based Fluids". *ACM Transactions on Graphics*. **32** (4): 1–12. doi:10.1145/2461912.2461984 (https://doi.org/10.1145%2F2461912.2461984). S2CID 611962 (https://api.semanticscholar.org/CorpusID:611962).
18. Akinci (2013). "Versatile Surface Tension and Adhesion for SPH Fluids SPH". *ACM Transactions on Graphics*. **32** (6): 1–8. CiteSeerX 10.1.1.462.8293 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.462.8293). doi:10.1145/2508363.2508395 (https://doi.org/10.1145%2F2508363.2508395). S2CID 12550964 (https://api.semanticscholar.org/CorpusID:12550964).
19. *Journal of Tribology* (2013). "Application of Smoothed Particle Hydrodynamics to Full-Film Lubrication". {{cite journal}}: Cite journal requires |journal= (help)
20. Mahdavi and Talebbeydokhti (2015). "A hybrid solid boundary treatment algorithm for smoothed particle hydrodynamics" (https://www.researchgate.net/publication/282870566). *Scientia Iranica, Transaction A, Civil Engineering*. **22** (4): 1457–1469.
21. *International Journal for Numerical Methods in Fluids* (2016). "Curvilinear smoothed particle hydrodynamics". *International Journal for Numerical Methods in Fluids*. **83** (2): 115–131. Bibcode:2017IJNMF..83..115T (https://ui.adsabs.harvard.edu/abs/2017IJNMF..83..115T). doi:10.1002/flid.4261 (https://doi.org/10.1002%2Fflid.4261). S2CID 124425780 (https://api.semanticscholar.org/CorpusID:124425780).
22. Price, Daniel J (2009). "Astrophysical Smooth Particle Hydrodynamics". *New Astron.rev*. **53** (4–6): 78–104. arXiv:0903.5075 (https://arxiv.org/abs/0903.5075). Bibcode:2009NewAR..53...78R (https://ui.adsabs.harvard.edu/abs/2009NewAR..53...78R). doi:10.1016/j.newar.2009.08.007 (https://doi.org/10.1016%2Fj.newar.2009.08.007). S2CID 129246 (https://api.semanticscholar.org/CorpusID:129246).
23. Rosswog, Stephan (2015). "SPH Methods in the Modelling of Compact Objects". *Living Rev Comput Astrophys*. **1** (1): 1. arXiv:1406.4224 (https://arxiv.org/abs/1406.4224). Bibcode:2015LRCA....1....1R (https://ui.adsabs.harvard.edu/abs/2015LRCA....1....1R). doi:10.1007/lrca-2015-1 (https://doi.org/10.1007%2Flrca-2015-1). S2CID 119119783 (https://api.semanticscholar.org/CorpusID:119119783).
24. Price, Daniel J; Rockefeller, Gabriel; Warren, Michael S (2006). "SNSPH: A Parallel 3-D Smoothed Particle Radiation Hydrodynamics Code". *Astrophys. J*. **643**: 292–305. arXiv:astro-ph/0512532 (https://arxiv.org/abs/astro-ph/0512532). doi:10.1086/501493 (https://doi.org/10.1086%2F501493). S2CID 16733573 (https://api.semanticscholar.org/CorpusID:16733573).

25. "Star Formation with Radiative Transfer" (<http://www.astro.ex.ac.uk/people/mbate/Cluster/clusterRT.html>).
26. Price, Daniel J. (2012-02-01). "Smoothed particle hydrodynamics and magnetohydrodynamics" (<https://www.sciencedirect.com/science/article/pii/S0021999110006753>). *Journal of Computational Physics*. Special Issue: Computational Plasma Physics. **231** (3): 759–794. arXiv:1012.1885 (<https://arxiv.org/abs/1012.1885>). doi:10.1016/j.jcp.2010.12.011 (<https://doi.org/10.1016%2Fj.jcp.2010.12.011>). ISSN 0021-9991 (<https://search.worldcat.org/issn/0021-9991>).
27. Libersky, L.D.; Petschek, A.G. (1990). "Smooth particle hydrodynamics with strength of materials". *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. Lecture Notes in Physics. Vol. 395. pp. 248–257. doi:10.1007/3-540-54960-9_58 (https://doi.org/10.1007%2F3-540-54960-9_58). ISBN 978-3-540-54960-4.
28. L.D. Libersky; A.G. Petschek; A.G. Carney; T.C. Hipp; J.R. Allahdadi; F.A. High (1993). "Strain Lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response". *J. Comput. Phys.* **109** (1): 67–75. Bibcode:1993JCoPh.109...67L (<https://ui.adsabs.harvard.edu/abs/1993JCoPh.109...67L>). doi:10.1006/jcph.1993.1199 (<https://doi.org/10.1006%2Fjcph.1993.1199>).
29. J.W. Swegle; D.A. Hicks; S.W. Attaway (1995). "Smooth particle hydrodynamics stability analysis". *J. Comput. Phys.* **116** (1): 123–134. Bibcode:1995JCoPh.116..123S (<https://ui.adsabs.harvard.edu/abs/1995JCoPh.116..123S>). doi:10.1006/jcph.1995.1010 (<https://doi.org/10.1006%2Fjcph.1995.1010>).
30. T. Belytschko; Y. Krongauz; J. Dolbow; C. Gerlach (1998). "On the completeness of meshfree particle methods". *Int. J. Numer. Methods Eng.* **43** (5): 785–819. Bibcode:1998IJNME..43..785B (<https://ui.adsabs.harvard.edu/abs/1998IJNME..43..785B>). CiteSeerX 10.1.1.28.491 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.491>). doi:10.1002/(sici)1097-0207(19981115)43:5<785::aid-nme420>3.0.co;2-9 (<https://doi.org/10.1002%2F%28sici%291097-0207%2819981115%2943%3A5%3C785%3A%3Aaid-nme420%3E3.0.co%3B2-9>).
31. W.K. Liu; S. Jun; Y.F. Zhang (1995). "Reproducing kernel particle methods". *Int. J. Numer. Methods Eng.* **20** (8–9): 1081–1106. Bibcode:1995IJNMF..20.1081L (<https://ui.adsabs.harvard.edu/abs/1995IJNMF..20.1081L>). doi:10.1002/flid.1650200824 (<https://doi.org/10.1002%2Ffid.1650200824>).
32. P.W. Randles; L.D. Libersky (1997). "Recent improvements in SPH modelling of hypervelocity impact". *Int. J. Impact Eng.* **20** (6–10): 525–532. doi:10.1016/s0734-743x(97)87441-6 (<https://doi.org/10.1016%2Fs0734-743x%2897%2987441-6>).
33. G.R. Johnson; S.R. Beissel (1996). "Normalized smoothing functions for SPH impact computations". *Int. J. Numer. Methods Eng.* **39** (16): 2725–2741. Bibcode:1996IJNME..39.2725J (<https://ui.adsabs.harvard.edu/abs/1996IJNME..39.2725J>). doi:10.1002/(sici)1097-0207(19960830)39:16<2725::aid-nme973>3.0.co;2-9 (<https://doi.org/10.1002%2F%28sici%291097-0207%2819960830%2939%3A16%3C2725%3A%3Aaid-nme973%3E3.0.co%3B2-9>).
34. C.T. Dyka; R.P. Ingel (1995). "An approach for tension instability in Smoothed Particle Hydrodynamics". *Comput. Struct.* **57** (4): 573–580. doi:10.1016/0045-7949(95)00059-p (<https://doi.org/10.1016%2F0045-7949%2895%2900059-p>).
35. C.T. Dyka; P.W. Randles; R.P. Ingel (1997). "Stress points for tension instability in SPH". *Int. J. Numer. Methods Eng.* **40** (13): 2325–2341. Bibcode:1997IJNME..40.2325D (<https://ui.adsabs.harvard.edu/abs/1997IJNME..40.2325D>). doi:10.1002/(sici)1097-0207(19970715)40:13<2325::aid-nme161>3.0.co;2-8 (<https://doi.org/10.1002%2F%28sici%291097-0207%2819970715%2940%3A13%3C2325%3A%3Aaid-nme161%3E3.0.co%3B2-8>).
36. P.W. Randles; L.D. Libersky (2000). "Normalized SPH with stress points". *Int. J. Numer. Methods Eng.* **48** (10): 1445–1462. Bibcode:2000IJNME..48.1445R (<https://ui.adsabs.harvard.edu/abs/2000IJNME..48.1445R>). doi:10.1002/1097-0207(20000810)48:10<1445::aid-nme831>3.0.co;2-9 (<https://doi.org/10.1002%2F1097-0207%2820000810%2948%3A10%3C1445%3A%3Aaid-nme831%3E3.0.co%3B2-9>).
37. T. Belytschko; Y. Guo; W.K. Liu; S.P. Xiao (2000). "A unified stability analysis of meshless particle methods". *Int. J. Numer. Methods Eng.* **48** (9): 1359–1400. Bibcode:2000IJNME..48.1359B (<https://ui.adsabs.harvard.edu/abs/2000IJNME..48.1359B>). doi:10.1002/1097-0207(20000730)48:9<1359::aid-nme829>3.0.co;2-u (<https://doi.org/10.1002%2F1097-0207%2820000730%2948%3A9%3C1359%3A%3Aaid-nme829%3E3.0.co%3B2-u>).
38. J. Bonet; S. Kulasegaram (2000). "Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations". *Int. J. Numer. Methods Eng.* **47** (6): 1189–1214. Bibcode:2000IJNME..47.1189B (<https://ui.adsabs.harvard.edu/abs/2000IJNME..47.1189B>). doi:10.1002/(sici)1097-0207(20000228)47:6<1189::aid-nme830>3.0.co;2-i (<https://doi.org/10.1002%2F%28sici%291097-0207%2820000228%2947%3A6%3C1189%3A%3Aaid-nme830%3E3.0.co%3B2-i>).
39. W. G. Hoover; C. G. Hoover (2001). "SPAM-based recipes for continuum simulations" (<https://zenodo.org/record/1262149>). *Computing in Science and Engineering*. **3** (2): 78–85. Bibcode:2001CSE.....3b..78H (<https://ui.adsabs.harvard.edu/abs/2001CSE.....3b..78H>). doi:10.1109/5992.909007 (<https://doi.org/10.1109%2F5992.909007>).
40. T. Rabczuk; J. Eibl; L. Stempniewski (2003). "Simulation of high velocity concrete fragmentation using SPH/MLSPH". *Int. J. Numer. Methods Eng.* **56** (10): 1421–1444. Bibcode:2003IJNME..56.1421R (<https://ui.adsabs.harvard.edu/abs/2003IJNME..56.1421R>). doi:10.1002/nme.617 (<https://doi.org/10.1002%2Fnm.617>). S2CID 119799557 (<https://api.semanticscholar.org/CorpusID:119799557>).

41. M.I. Herreros; M. Mabssout (2011). "A two-steps time discretization scheme using the SPH method for shock wave propagation". *Comput. Methods Appl. Mech. Engrg.* **200** (21–22): 1833–1845. Bibcode:2011CMAME.200.1833H (<https://ui.adsabs.harvard.edu/abs/2011CMAME.200.1833H>). doi:10.1016/j.cma.2011.02.006 (<https://doi.org/10.1016%2Fj.cma.2011.02.006>).
42. S. Yashiro; T. Okabe (2015). "Smoothed particle hydrodynamics in a generalized coordinate system with a finite-deformation constitutive model". *Int. J. Numer. Methods Eng.* **103** (11): 781–797. Bibcode:2015IJNME.103..781Y (<https://ui.adsabs.harvard.edu/abs/2015IJNME.103..781Y>). doi:10.1002/nme.4906 (<https://doi.org/10.1002%2Fnme.4906>). hdl:2324/4476056 (<https://hdl.handle.net/2324%2F4476056>). S2CID 121935150 (<https://api.semanticscholar.org/CorpusID:121935150>).
43. N.J. Quinlan; M. Basa; M. Lastiwka (2006). "Truncation error in mesh-free particle methods" (https://aran.library.nuigalway.ie/bitstream/10379/1170/3/truncationError_Quinlan2005_preprint.pdf) (PDF). *International Journal for Numerical Methods in Engineering.* **66** (13): 2064–2085. Bibcode:2006IJNME..66.2064Q (<https://ui.adsabs.harvard.edu/abs/2006IJNME..66.2064Q>). doi:10.1002/nme.1617 (<https://doi.org/10.1002%2Fnme.1617>). hdl:10379/1170 (<https://hdl.handle.net/10379%2F1170>). S2CID 53701216 (<https://api.semanticscholar.org/CorpusID:53701216>).
44. H. Wendland (1995). "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree". *Advances in Computational Mathematics.* **4** (4): 389–396. doi:10.1007/BF02123482 (<https://doi.org/10.1007%2FBF02123482>). S2CID 36452865 (<https://api.semanticscholar.org/CorpusID:36452865>).
45. A. Mayrhofer; B.D. Rogers; D. Violeau; M. Ferrand (2013). "Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions". *Computer Physics Communications.* **184** (11): 2515–2527. arXiv:1304.3692 (<https://arxiv.org/abs/1304.3692>). Bibcode:2013CoPhC.184.2515M (<https://ui.adsabs.harvard.edu/abs/2013CoPhC.184.2515M>). CiteSeerX 10.1.1.770.4985 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.770.4985>). doi:10.1016/j.cpc.2013.07.004 (<https://doi.org/10.1016%2Fj.cpc.2013.07.004>). S2CID 35008128 (<https://api.semanticscholar.org/CorpusID:35008128>).
46. J. Bonet; T.S. Lok (1999). "Variational and momentum preservation aspects of Smoothed Particle Hydrodynamics formulations". *Computers Methods in Applied Mechanical Engineering.* **180** (1–2): 97–115. Bibcode:1999CMAME.180...97B (<https://ui.adsabs.harvard.edu/abs/1999CMAME.180...97B>). doi:10.1016/S0045-7825(99)00051-1 (<https://doi.org/10.1016%2FS0045-7825%2899%2900051-1>).
47. J.J. Monaghan (2005). "Smoothed particle hydrodynamics". *Reports on Progress in Physics.* **68** (8): 1703–1759. Bibcode:2005RPPh...68.1703M (<https://ui.adsabs.harvard.edu/abs/2005RPPh...68.1703M>). doi:10.1088/0034-4885/68/8/R01 (<https://doi.org/10.1088%2F0034-4885%2F68%2F8%2FR01>). S2CID 5987481 (<https://api.semanticscholar.org/CorpusID:5987481>).
48. E. Hairer; C. Lubich; G. Wanner (2006). *Geometric Numerical Integration* (<https://www.springer.com/us/book/9783540306634>). Springer. ISBN 978-3-540-30666-5.
49. Andrea Colagrossi; Matteo Antuono; David Le Touzè (2009). "Theoretical considerations on the free-surface role in the smoothed-particle-hydrodynamics model". *Physical Review E.* **79** (5): 056701. Bibcode:2009PhRvE..79e6701C (<https://ui.adsabs.harvard.edu/abs/2009PhRvE..79e6701C>). doi:10.1103/PhysRevE.79.056701 (<https://doi.org/10.1103%2FPhysRevE.79.056701>). PMID 19518587 (<https://pubmed.ncbi.nlm.nih.gov/19518587/>).
50. Benjamin Bouscasse; Andrea Colagrossi; Salvatore Marrone; Matteo Antuono (2013). "Nonlinear water wave interaction with floating bodies in SPH". *Journal of Fluids and Structures.* **42**: 112–129. Bibcode:2013JFS....42..112B (<https://ui.adsabs.harvard.edu/abs/2013JFS....42..112B>). doi:10.1016/j.jfluidstructs.2013.05.010 (<https://doi.org/10.1016%2Fj.jfluidstructs.2013.05.010>).
51. Fabricio Macià; Matteo Antuono; Leo M González; Andrea Colagrossi (2011). "Theoretical analysis of the no-slip boundary condition enforcement in SPH methods" (<https://doi.org/10.1143%2FPTP.125.1091>). *Progress of Theoretical Physics.* **125** (6): 1091–1121. Bibcode:2011PTPh.125.1091M (<https://ui.adsabs.harvard.edu/abs/2011PTPh.125.1091M>). doi:10.1143/PTP.125.1091 (<https://doi.org/10.1143%2FPTP.125.1091>).
52. Jose Luis Cercos-Pita; Matteo Antuono; Andrea Colagrossi; Antonio Souto (2017). "SPH energy conservation for fluid–solid interactions". *Computer Methods in Applied Mechanics and Engineering.* **317**: 771–791. Bibcode:2017CMAME.317..771C (<https://ui.adsabs.harvard.edu/abs/2017CMAME.317..771C>). doi:10.1016/j.cma.2016.12.037 (<https://doi.org/10.1016%2Fj.cma.2016.12.037>).
53. J. Campbell; R. Vignjevic; L. Libersky (2000). "A contact algorithm for smoothed particle hydrodynamics". *Computer Methods in Applied Mechanics and Engineering.* **184** (1): 49–65. Bibcode:2000CMAME.184...49C (<https://ui.adsabs.harvard.edu/abs/2000CMAME.184...49C>). doi:10.1016/S0045-7825(99)00442-9 ([https://doi.org/10.1016%2FS0045-7825\(99\)00442-9](https://doi.org/10.1016%2FS0045-7825(99)00442-9)).
54. M. Ferrand, D.R. Laurence, B.D. Rogers, D. Violeau, C. Kassiotis (2013). "Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method" (<https://hal-enpc.archives-ouvertes.fr/hal-00691603/document>). *International Journal for Numerical Methods in Fluids.* **71** (4). *Int. J. Numer. Meth. Fluids:* 446–472. Bibcode:2013IJNMF..71..446F (<https://ui.adsabs.harvard.edu/abs/2013IJNMF..71..446F>). doi:10.1002/flid.3666 (<https://doi.org/10.1002%2Ffid.3666>). S2CID 124465492 (<https://api.semanticscholar.org/CorpusID:124465492>).
55. H. R. Cole (1948). *Underwater Explosions* (<https://archive.org/details/underwaterexplos00cole>). Princeton, New Jersey: Princeton University Press.

56. D. Molteni, A. Colagrossi (2009). "A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH". *Computer Physics Communications*. **180** (6): 861–872. Bibcode:2009CoPhC.180..861M (<https://ui.adsabs.harvard.edu/abs/2009CoPhC.180..861M>). doi:10.1016/j.cpc.2008.12.004 (<https://doi.org/10.1016%2Fj.cpc.2008.12.004>).
57. Colagrossi, Andrea; Landrini, Maurizio (2003). "Numerical simulation of interfacial flows by smoothed particle hydrodynamics". *Journal of Computational Physics*. **191** (2): 448–475. Bibcode:2003JCoPh.191..448C (<https://ui.adsabs.harvard.edu/abs/2003JCoPh.191..448C>). doi:10.1016/S0021-9991(03)00324-3 (<https://doi.org/10.1016%2FS0021-9991%2803%2900324-3>).
58. Randall J. LeVeque (2007). *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Siam.
59. A. Ferrari, M. Dumbser, E. Toro, A. Armanini (2009). "A new 3D parallel SPH scheme for free surface flows". *Computers & Fluids*. **38** (6). Elsevier: 1203–1217. doi:10.1016/j.compfluid.2008.11.012 (<https://doi.org/10.1016%2Fj.compfluid.2008.11.012>).
60. Fatehi, R and Manzari, MT (2011). "A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics". *International Journal for Numerical Methods in Fluids*. **67** (9). Wiley Online Library: 1100–1114. Bibcode:2011IJNMF.67.1100F (<https://ui.adsabs.harvard.edu/abs/2011IJNMF.67.1100F>). doi:10.1002/flid.2406 (<https://doi.org/10.1002%2Fflid.2406>). S2CID 121381641 (<https://api.semanticscholar.org/CorpusID:121381641>).
61. M. Antuono, A. Colagrossi, S. Marrone, D. Molteni (2010). "Free-surface flows solved by means of SPH schemes with numerical diffusive terms". *Computer Physics Communications*. **181** (3). Elsevier: 532–549. Bibcode:2010CoPhC.181..532A (<https://ui.adsabs.harvard.edu/abs/2010CoPhC.181..532A>). doi:10.1016/j.cpc.2009.11.002 (<https://doi.org/10.1016%2Fj.cpc.2009.11.002>).
62. M. Antuono, A. Colagrossi, S. Marrone (2012). "Numerical diffusive terms in weakly-compressible SPH schemes". *Computer Physics Communications*. **183** (12). Elsevier: 2570–2580. Bibcode:2012CoPhC.183.2570A (<https://ui.adsabs.harvard.edu/abs/2012CoPhC.183.2570A>). doi:10.1016/j.cpc.2012.07.006 (<https://doi.org/10.1016%2Fj.cpc.2012.07.006>).
63. Antuono Matteo and Marrone S and Colagrossi A and Bouscasse B (2015). "Energy balance in the δ -SPH scheme". *Computer Methods in Applied Mechanics and Engineering*. **289**. Elsevier: 209–226. Bibcode:2015CMAME.289..209A (<https://ui.adsabs.harvard.edu/abs/2015CMAME.289..209A>). doi:10.1016/j.cma.2015.02.004 (<https://doi.org/10.1016%2Fj.cma.2015.02.004>).
64. JP. Vila (1999). "On particle weighted methods and smooth particle hydrodynamics". *Mathematical Models and Methods in Applied Sciences*. **9** (2). World Scientific: 161–209. doi:10.1142/S0218202599000117 (<https://doi.org/10.1142%2FS0218202599000117>).
65. Marongiu Jean-Christophe and Leboeuf Francis and Caro Joëlle and Parkinson Etienne (2010). "Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method" (<https://hal.archives-ouvertes.fr/hal-00566051/file/marongiu2010.pdf>) (PDF). *Journal of Hydraulic Research*. **48** (S1). Taylor & Francis: 40–49. doi:10.1080/00221686.2010.9641244 (<https://doi.org/10.1080%2F00221686.2010.9641244>). S2CID 121493014 (<https://api.semanticscholar.org/CorpusID:121493014>).
66. De Lefle, Matthieu (2011). *Modelisation d'écoulements visqueux par methode SPH en vue d'application à l'hydrodynamique navale*. PhD Thesis, Ecole centrale de Nantes.
67. Chi Zhang and Xiangyu Hu and Nikolaus Adams (2017). "A weakly compressible SPH method based on a low-dissipation Riemann solver". *Journal of Computational Physics*. **335**: 605–620. Bibcode:2017JCoPh.335..605Z (<https://ui.adsabs.harvard.edu/abs/2017JCoPh.335..605Z>). doi:10.1016/j.jcp.2017.01.027 (<https://doi.org/10.1016%2Fj.jcp.2017.01.027>).
68. Monaghan, Joseph J.; Gingold, Robert A. (1983). "Shock Simulation by the Particle Method". *Journal of Computational Physics*. **52** (2): 347–389. Bibcode:1983JCoPh..52..374M (<https://ui.adsabs.harvard.edu/abs/1983JCoPh..52..374M>). doi:10.1016/0021-9991(83)90036-0 (<https://doi.org/10.1016%2F0021-9991%2883%290036-0>).
69. J. P. Morris; P. J. Fox; Y. Zhu (1997). "Modeling Low Reynolds Number Incompressible Flows Using SPH". *Journal of Computational Physics*. **136** (1): 214–226. Bibcode:1997JCoPh.136..214M (<https://ui.adsabs.harvard.edu/abs/1997JCoPh.136..214M>). doi:10.1006/jcph.1997.5776 (<https://doi.org/10.1006%2Fjcph.1997.5776>).
70. Marios D. Dikaiakos; Joachim Stadel, *PKDGRAV The Parallel k-D Tree Gravity Code* (<http://www-hpcc.astro.washington.edu/faculty/trq/brandon/pkdgrav.html>), retrieved February 1, 2017
71. Vignjevic, Rade; Reveles, Juan R.; Campbell, James (2006). "SPH in a total Lagrangian formalism". *Computer Modeling in Engineering and Sciences*. **44**: 181–198.
72. Han, Luhui; Hu, Xiangyu (2018). "SPH modeling of fluid-structure interaction". *Journal of Hydrodynamics*. **30** (1): 62–69. Bibcode:2018JHyDy..30...62H (<https://ui.adsabs.harvard.edu/abs/2018JHyDy..30...62H>). doi:10.1007/s42241-018-0006-9 (<https://doi.org/10.1007%2FS42241-018-0006-9>). S2CID 125369012 (<https://api.semanticscholar.org/CorpusID:125369012>).
73. Chi Zhang; Massoud Rezavand; Xiangyu Hu (2020). "A multi-resolution SPH method for fluid-structure interactions". *Journal of Computational Physics*. **429**: 110028. arXiv:1911.13255 (<https://arxiv.org/abs/1911.13255>). doi:10.1016/j.jcp.2020.110028 (<https://doi.org/10.1016%2Fj.jcp.2020.110028>). ISSN 0021-9991 (<https://search.worldcat.org/issn/0021-9991>). S2CID 208513116 (<https://api.semanticscholar.org/CorpusID:208513116>).

Further reading

- Hoover, W. G. (2006); *Smooth Particle Applied Mechanics: The State of the Art*, World Scientific.
- Stellingwerf, R. F.; Wingate, C. A.; "Impact Modelling with SPH" (<http://adsabs.harvard.edu/abs/1994MmSAI..65.1117S>), *Memorie della Societa Astronomia Italiana*, Vol. 65, p. 1117 (1994).
- Amada, T.; Imura, M.; Yasumuro, Y.; Manabe, Y.; and Chihara, K. (2004); "Particle-based fluid simulation on GPU", in *Proceedings of ACM Workshop on General-purpose Computing on Graphics Processors* (August, 2004, Los Angeles, California).
- Desbrun, M.; and Cani, M.-P. (1996). "Smoothed Particles: a new paradigm for animating highly deformable bodies" in *Proceedings of Eurographics Workshop on Computer Animation and Simulation* (August 1996, Poitiers, France).
- Hegeman, K.; Carr, N. A.; and Miller, G. S. P.; "Particle-based fluid simulation on the GPU", in *Proceedings of International Conference on Computational Science* (Reading, UK, May 2006), *Lecture Notes in Computer Science* v. 3994/2006 (Springer-Verlag).
- Kelager, M. (2006) *Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics*, MSc Thesis, Univ. Copenhagen.
- Kolb, A.; and Cuntz, N. (2005); "Dynamic particle coupling for GPU-based fluid simulation", in *Proceedings of the 18th Symposium on Simulation Techniques* (2005) pp. 722-727.
- Liu, G. R.; and Liu, M. B.; *Smoothed Particle Hydrodynamics: a meshfree particle method*, Singapore: World Scientific (2003).
- Monaghan, Joseph J. (1992). "Smoothed Particle Hydrodynamics", *Annual Review of Astronomy and Astrophysics* (1992). 30 : 543-74.
- Muller, M.; Charypar, D.; and Gross, M.; "Particle-based Fluid Simulation for Interactive Applications", in Breen, D; and Lin, M. (eds.), *Proceedings of Eurographics/SIGGRAPH Symposium on Computer Animation* (2003).
- Vesterlund, M.; *Simulation and Rendering of a Viscous Fluid Using Smoothed Particle Hydrodynamics*, MSc Thesis, Umea University, Sweden.
- Violeau, D.; *Fluid Mechanics and the SPH method*, Oxford University Press (2012).

External links

- First large simulation of star formation using SPH (<http://www.ukaff.ac.uk/starcluster>)
- SPHERIC (SPH rEsearch and engineeriNg International Community) (<http://spheric-sph.org/>)
- ITVO (http://ia2.oats.inaf.it/index.php?option=com_content&view=section&id=12&Itemid=71) is the website of The Italian Theoretical Virtual Observatory created to query a database of numerical simulation archive.
- SPHC Image Gallery (<http://www.stellingwerf.com/rfs-bin/index.cgi?action=PageView&id=33>) depicts a wide variety of test cases, experimental validations, and commercial applications of the SPH code SPHC.
- A derivation of the SPH model starting from Navier-Stokes equations (<http://alumnus.caltech.edu/~heirich/resources/SmoothedParticleHydrodynamics.pdf>)

Software

- Algodoo is a 2D simulation framework for education using SPH (<http://www.algodoo.com/>)
- AQUAgpusph (<http://canal.etsin.upm.es/aquagpusph>) is the free (GPLv3) SPH of the researchers, by the researchers, for the researchers
- dive solutions (<https://dive-solutions.de>) is a commercial web-based SPH engineering software for CFD purposes
- DualSPHysics (<http://www.dual.sphysics.org>) is a mostly open source SPH code based on SPHysics and using GPU computing. The open source components are available under the LGPL.
- FLUIDS v.1 (<http://www.rchoetzlein.com/eng/>) is a simple, open source (Zlib), real-time 3D SPH implementation in C++ for liquids for CPU and GPU.
- Fluidix (<https://www.fluidix.ca>) is a GPU-based particle simulation API available from OneZero Software
- GADGET [1] (<http://www.mpa-garching.mpg.de/galform/gadget/>) is a freely available (GPL) code for cosmological N-body/SPH simulations
- GPUSPH (<http://www.gpusph.org>) SPH simulator with viscosity (GPLv3)
- Pasimodo (http://www.itm.uni-stuttgart.de/research/pasimodo/pasimodo_en.php) is a program package for particle-based simulation methods, e.g. SPH
- LAMMPS (<https://www.lammps.org/>) is a massively parallel, open-source classical molecular dynamics code

that can perform SPH simulations

- **Physics Abstraction Layer** (<http://pal.sourceforge.net/>) is an open source abstraction system that supports real time physics engines with SPH support
 - **PreonLab** (<https://www.fifty2.eu/preonlab/>) is a commercial engineering software developed by **FIFTY2 Technology** (<https://www.fifty2.eu/>) implementing an implicit SPH method
 - **Punto** (<http://sourceforge.net/projects/punto/>) is a freely available visualisation tool for particle simulations
 - **pysph** (<https://github.com/pypr/pysph>) Open Source Framework for Smoothed Particle Hydrodynamics in Python (New BSD License)
 - **Py-SPHViewer** (<http://alejandrobll.github.io/py-sphviewer/>) Open Source python visualisation tool for Smoothed Particle Hydrodynamics simulations.^[1]
 - **RealFlow** (<https://realflow.com/>) Commercial SPH solver for the cinema industry.
 - **RheoCube** (<https://rheocube.com>) is a commercial SaaS product by **Lorenz Research** (<https://lorenz.re>) for the study and prediction of complex-fluid rheology and stability
 - **SimPARTIX** (<http://www.simpartix.com>) is a commercial simulation package for SPH and **Discrete element method** (DEM) simulations from **Fraunhofer IWM**
 - **SPH-flow** (<https://www.nextflow-software.com/sphflow/>)
 - **SPHERA** (<https://github.com/AndreaAmicarelliRSE/SPHERA>)
 - **SPHinXsys** (<http://www.sphinxsys.org>) is an open source multi-physics, multi-resolution SPH library. It provides C++ APIs for physical accurate simulation and aims to model coupled industrial dynamic systems including fluid, solid, multi-body dynamics and beyond.
 - **SPHysics** (<http://www.sphysics.org/>) is an open source SPH implementation in Fortran
 - **SPLASH** (<https://web.archive.org/web/20071113233849/http://www.astro.ex.ac.uk/people/dprice/splash/>) is an open source (GPL) visualisation tool for SPH simulations
 - **SYMPLER** (<http://sympler.org>): A freeware SYMBolic ParticLE simulaToR from the University of Freiburg.
 - **Nauticle** (<https://bitbucket.org/nauticleproject/nauticle/src/master/>) is a general-purpose computational tool for particle-based numerical methods.
 - **NDYNAMICS** (<http://www.centroidlab.com/neutrinodynamics>) is a commercial fluid simulation software based on implicit SPH developed by **CENTROID LAB** (<https://www.centroidlab.com>) currently used for internal/external flooding/nuclear/chemical engineering applications.
1. Benitez-Llambay, Alejandro (2015-07-28), "Py-Sphviewer: Py-Sphviewer V1.0.0" (<https://zenodo.org/record/21703>), *Zenodo*, Bibcode:2015zndo.....21703B (<https://ui.adsabs.harvard.edu/abs/2015zndo.....21703B>), doi:10.5281/zenodo.21703 (<https://doi.org/10.5281%2Fzenodo.21703>), retrieved 2022-03-30

Retrieved from "https://en.wikipedia.org/w/index.php?title=Smoothed-particle_hydrodynamics&oldid=1285712988"