Smoothed Particles: A new paradigm for animating highly deformable bodies

Mathieu Desbrun   Marie-Paule Gascuel

iMAGIS* - GRAVIR / IMAG
BP 53, F-38041 Grenoble cedex 09, France

Abstract

This paper presents a new formalism for simulating highly deformable bodies with a particle system. Smoothed particles represent sample points that enable the approximation of the values and derivatives of local physical quantities inside a medium. They ensure valid and stable simulation of state equations that describe the physical behavior of the material.

We extend the initial formalism, first introduced for simulating cosmological fluids, to the animation of inelastic bodies with a wide range of stiffness and viscosity. We show that the smoothed particles paradigm leads to a coherent definition of the object's surface as an iso-surface of the mass density function. Implementation issues are discussed, including an efficient integration scheme using individually adapted time steps to integrate particle motion. Animation requires a linear complexity in the number of particles, offering reasonable time and memory use.

1 Introduction

Modeling and simulating deformable bodies has attracted a lot of attention in Computer Graphics. Elastic materials that recover their rest shape after any deformation are generally simulated with finite-difference [TPBF87] or finite-element [GMTT89, PW89] methods. Nevertheless, these approaches, which approximate the body by a mesh of nodes of fixed topology, are not adapted to the animation of substances able to undergo large inelastic deformations. In this case, the use of particle systems is more appropriate. These systems are unstructured in the sense that interactions between point masses do not depend on a specified graph of connections, but on distance. Consequently, particle systems seem the easiest approach for modeling large changes in shape and in topology.

*iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble, and Université Joseph Fourier.

1.1 Previous approaches

Particle systems were first defined in Computer Graphics as sets of moving points, without interaction between them [Ree83]. More recently, particles systems have been widely used for simulating inelastic deformations and even fluids [MP89, TPF89, LJR+91, Ton91, DG95, LP95, LHVD95, GLG95]. Most methods developed so far use different simplified versions of the same attraction-repulsion force for modeling interactions between particles. This force derives from the Lennard-Jones potential, proposed for modeling pairwise microscopic interactions between atoms in a liquid. Animating n particles amounts to calculating at each time step the n^2 interaction forces, and then integrating the equations of motion for each particle to obtain its speed and position. To allow better performance, interaction forces are clamped to zero at a cutoff radius, reducing the number of interacting neighbors.

Despite many advantages such as the simplicity of implementation, the practical use of particle systems gives rise to a variety of problems. First, the parameters of Lennard-Jones interaction forces are not easy to manipulate. Finding values that will result in a desired global behavior is quite difficult. This may be related to the fact that Lennard-Jones forces come from microscopic observations, and here are applied on a totally different scale.

Time integration is also an important source of problems. As no stability criterion is provided, the integration time step is very difficult to handle. Even at rest positions (when particles are supposed to freeze), forces must be either integrated carefully or modified to avoid oscillations [LP95]. For all these reasons, small time steps seem inescapable.

A last problem, quite important in Computer Graphics applications, is the lack of definition of the surface surrounding the object. Providing such a surface is essential for interfacing particles with other models. A solution for computing a smooth surface from particles consists in using an iso-surface of some "implicit function" controlled by the particles [MP89, Ton91]. The resulting implicit surface can be used for collision and contact processing between the body and its environment [Gas93, DG95]. However, coating particles with an arbitrary implicit function introduces large volume variations. This problem should not occur if a coherent definition of the surface can be found since particles mimic a material that is supposed to keep its volume nearly constant.

1.2 Overview

This paper presents a new paradigm for simulating highly deformable substances with particle systems. We extend the Smoothed Particle Hydrodynamics (SPH) approach used by physicists for cosmological fluid simulation. The smoothed particles we define can be viewed either as matter elements, or sample points scattered in a soft substance. Each of them represents a small volume of inelastic material that moves over time. In practice, smoothed particles are used to approximate the values and derivatives of continuous physical quantities, such as local mass density or pressure, that need to be computed during the simulation. Smoothed particles ensure valid and stable simulation of a state equation describing the physical behavior of the material. We also use them for defining the surface of the substance in a coherent way using the level sets of the mass density function. Implementation issues are discussed, and, in particular, an efficient integration scheme with adapted time steps for each particle is presented.

2 Smoothed Particle Hydrodynamics

The Smoothed Particle Hydrodynamics (SPH) formalism was introduced by physicists for accurate simulation of fluid dynamics [Mon92]. Simulating a fluid consists in computing the variations of continuous functions such as mass density, speed, pressure, or temperature over space and time. Standard finite element techniques in hydrodynamics use an Eulerian approach: they consist of dividing space into a fixed grid of voxels, and then studying what flows in or out of each voxel. However, this kind of approach requires the division of huge empty volumes and is not intuitive for flows. SPH belongs to an alternative approach, called the Lagrangian approach, that consists of following the evolution of selected fluid elements over space and time. We describe in this section the fundamentals of this formalism as it provides sound bases for simulation, with simple implementation.

2.1 Discrete formulation of continuous fields

In SPH, the fluid is sampled by a set of elements called *particles*. A particle j has a fixed mass m_j, a position $\mathbf{r}_j$, a velocity $\mathbf{v}_j$, and a mass density ρ_j depending on the local density of particles. As a sample point, it can also carry physical field values like pressure or temperature. Then in a way very similar to Monte-Carlo techniques[PTVF92], these fields and their derivatives can be approximated by a discrete sum. To achieve this, particles are supposed to be smeared out in space according to a smoothing kernel W_h. This normalized kernel (i.e., the value of its integral is one) gives the spatial mass distribution profile over a smoothing length h. Then the smoothed values and derivatives of a continuous field f known only at particle locations can be approximated by:

$$< f(\mathbf{r}) > \quad = \quad \sum_j m_j \frac{f_j}{\rho_j} W_h(\mathbf{r} - \mathbf{r}_j) \tag{1}$$

$$< \nabla f(\mathbf{r}) > \quad = \quad \sum_j m_j \frac{f_j}{\rho_j} \nabla W_h(\mathbf{r} - \mathbf{r}_j) \tag{2}$$

where f_j denotes $f(\mathbf{r}_j)$, i.e. the value of f at particle j. Mathematical justification of the integral expressions we approximate by discrete sums can be found in Appendix A. These equations are the basis of the SPH formalism.

An immediate result gives a smoothed value of the mass density, defined everywhere by:

$$< \rho(\mathbf{r}) > = \sum_j m_j W_h(\mathbf{r} - \mathbf{r}_j). \tag{3}$$

2.2 Pressure forces

Equations (1) and (2) can be applied to derive smoothed versions of the hydrodynamical conservation laws, producing simple equations of motion for the particles. From here on, we will no longer distinguish between a function and its smoothed approximation. Thus the density ρ_i of a particle i is computed by the equation (3).

If the pressure P_i is known at each particle i, forces due to differences of pressure can be approximated. As these forces are locally proportional to the gradient of the pressure, Appendix B demonstrates that a symmetric expression of the pressure force on particle i can be written:

$$\mathbf{F}_i = -m_i \sum_{j \neq i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_h^{ij} \tag{4}$$

where the notation $\nabla_i W_h^{ij}$ denotes the gradient of $W_h(\mathbf{r}_i - \mathbf{r}_j)$.

The value of pressure in equation (4) is computed from a state equation describing the simulated fluid. For instance, an ideal gas satisfies $PV = k$ where V is the volume of the fluid and k a given constant. It results in purely repulsive forces between particles, accounting for the fact that an ideal gas tends to expand in free space.

2.3 Viscosity

SPH most commonly expresses viscosity by adding a damping force term to the equation of motion of a particle:

$$\mathbf{D}_i = -m_i \sum_{j \neq i} m_j \Pi_{ij} \nabla_i W_h^{ij} \tag{5}$$

where:

$$\Pi_{ij} = \begin{cases} \frac{-c\mu_{ij} + 2\mu_{ij}^2}{\overline{\rho}_{ij}} & \text{if } \mu_{ij} < 0 \\ 0 & \text{if } \mu_{ij} \geq 0 \end{cases} \qquad \mu_{ij} = h \frac{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{\mathbf{r}_{ij}^2 + h^2/100}$$

$$\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j \qquad \mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j \qquad \overline{\rho}_{ij} = (\rho_i + \rho_j)/2$$

The constant c is the *speed of sound* of the simulated fluid, which represents the fastest velocity of a wave front propagating in that medium. It indicates at which speed a deformation will be transmitted to the whole material. But it does not mean that particles cannot go faster than c: if external forces make them move faster, a shock wave occurs.

The first term in the expression of Π_{ij} is analogous to a shear and bulk viscosity. The second one, comparable to the Von Neumann-Richtmyer artificial viscosity used in grid-based methods, prevents particle interpenetration at high speed. We can note that the viscosity vanishes for rigid body motion and conserves linear and angular momenta as it depends on relative speeds of particle pairs.

3 Simulating highly deformable bodies with smoothed particles

The SPH approach provides a robust and reliable tool for fluid simulation, and has been widely used for simulating complex phenomena in astrophysics. However, although the smoothed particles paradigm is general, SPH does not directly apply to Computer Graphics. We no longer want to accurately simulate fluids, but rather to animate a wide range of inelastic deformable bodies. Several additions and modifications to the initial approach need to be defined.

3.1 Interaction Force Design

An important aspect of smoothed particles is that it derives local forces between pairs of neighbors particles from a global state equation. Such an equation describes how the physical variables evolve in the simulated material.

Pressure and cohesion forces

In SPH, the expression of pressure used in equation (4) resulted in positive, i.e., purely repulsive, forces expressing the natural expansion of the fluid. In astrophysics applications, pressure forces were often combined with gravitational forces balancing the expansion phenomenon.

In contrast, we would like to animate materials with constant density at rest. Consequently, the material should exhibit some internal cohesion, resulting in attraction-repulsion forces as in the Lennard-Jones model. To keep density at ρ_0, we replace the ideal gas state equation by:

$$P = k(\rho - \rho_0) \tag{6}$$

which is another expression of: $(P + P_0)V = k$, where $V = 1/\rho$ is the volume per unit mass, and $P_0 = k\rho_0$. Equation (6), designed to maintain density close to a constant value, has a double advantage. First, if particles have the same mass they will tend to be evenly distributed inside the object. This is essential since we are using them as sample points for approximating continuous functions. Moreover, constant density results in a constant volume. The material will then tend to naturally come back to its initial volume after a deformation.

Replacing P by its value in equation (4) leads to:

$$\mathbf{F}_i = -k m_i \left[\frac{(\rho_i - \rho_0)}{\rho_i^2} \sum_{j \neq i} m_j \nabla_i W_h^{ij} \; + \; \sum_{j \neq i} m_j \frac{(\rho_j - \rho_0)}{\rho_j^2} \nabla_i W_h^{ij} \right] \tag{7}$$

Interpretation

Equation (7) can be interpreted in the following way:

- The first term is a *density gradient descent*, that tends to minimize the difference between current and desired densities.

- The second term is a symmetry term that ensures the action-reaction principle.

The parameter k determines the strengh of the density recovery. It plays the same role than a stiffness parameter in a standard particle system. A large k will simulate a stiff material while a small k models a soft one, that recovers its rest density slowly after a compression or an expansion.

We combine the new pressure/cohesion force $\mathbf{F}_i$ with forces due to viscosity, as was done in the original SPH formalism (section 2.3).

3.2 Choice of a smoothing kernel

The choice of the smoothing kernel W_h is very important: if the particles are considered as sample points, we can compare the kernel with the difference scheme used in a finite-differences method, since it gives a way to approximate values and derivatives of various functions. And if particles are thought of as small matter elements, the kernel is related to the "extent" of a particle in space. In particular, the smoothing length h defining the support of the kernel (see Figure 1) gives the radius of influence of interaction forces created by a particle. Different behaviors can be obtained by tuning h. A small value will create very local interactions so the body will separate more easily into pieces.

We can also note that the kernel's support is related to the computational complexity of the simulation: if particles are well distributed, the mean number m of them in a sphere of radius $2h$ gives the mean number of interactions to compute for each particle.

Previous studies on the SPH formalism [Mon92] have shown that good kernels have a Fourier transform that falls rapidly with wave number. Most researchers use the spline Gaussian kernel:

$$W_h(\mathbf{r}) = \frac{1}{\pi h} \left\{ \begin{array}{cl} 1 - \frac{3}{2}(\frac{r}{h})^2 + \frac{3}{4}(\frac{r}{h})^3 & \text{if } 0 \le r \le h \\ \frac{1}{4}(2 - \frac{r}{h})^3 & \text{if } h \le r \le 2h \\ 0 & \text{if } r > 2h \end{array} \right.$$

This kernel, depicted in Figure 1, mimics the Gaussian bell curve and has a compact support, which implies both a finite radius of influence, and simpler computation. However, difficulties arise when the spline kernel is used for evaluating interaction forces between particles. Forces are more and more attenuated when two particles get closer to each other, since ∇W_h, given in Figure 1, is a factor in the expression of forces. So the spline kernel gives rise to clustering. However, since astrophysicists typically combine pressure forces with gravitational forces, this artifact is turned into feature in their applications.

For our application, we cannot use a kernel that will induce clustering between particles. This would be the opposite of our wish for constant density and regular sampling. Thus, we propose another normalized kernel, depicted in Figure 2:

$$W_h(\mathbf{r}) = \frac{15}{\pi(4h)^3} \left\{ \begin{array}{cl} (2 - \frac{r}{h})^3 & \text{if } 0 \le r \le 2h \\ 0 & \text{if } r > 2h \end{array} \right.$$

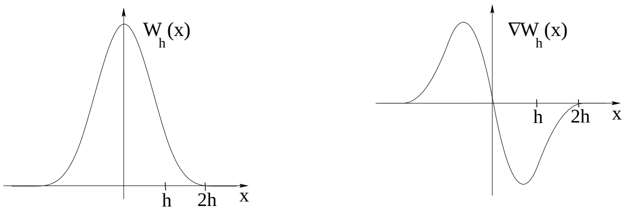

Figure 1: The spline kernel and its first derivative in one dimension.

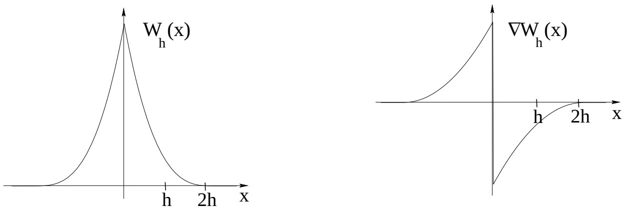

Figure 2: Our alternative kernel and its derivative; clustering is avoided.

This kernel keeps the advantages mentioned above while being better designed to handle nearby particles. As a result, the attraction/repulsion force

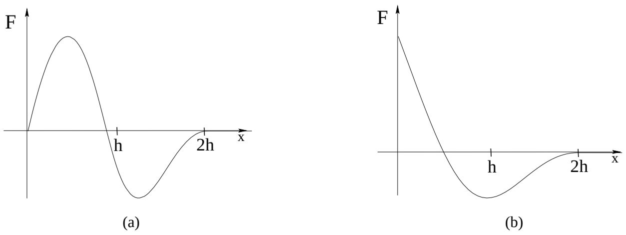

Figure 3: Comparison of pressure/cohesion forces between two particles:
(a) with the spline kernel (b) with the new kernel.

generated by equation (7) between a pair of particles looks satisfactory since it is very similar to the usual Lennard-Jones attraction/repulsion force, as shown in Figure 3.

3.3 Results

Testing the model in 2D is interesting since it enables to observe the evolution of the mass density in space and time. Density values, that can be computed everywhere, are displayed in shades of grey.

Figure 4 exhibits snapshots from a simple animation where 80 smoothed particles are used. In this example, parameters values are: $k = 10, c = 2$. h is constrained by ρ_0 so that there are approximately 10 particles in the radius of influence of each particle. Other experiments have proven the relevancy of physical parameters c and k which represent viscosity and stiffness, respectively. We also obtain good stability results.

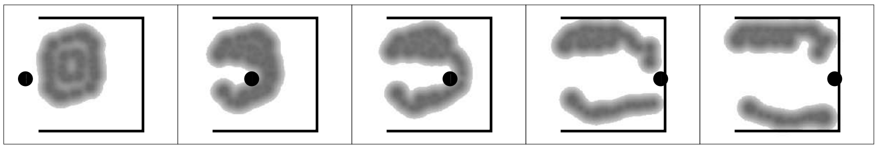

Figure 4: Snapshots from a 2D animation, where mass density is represented in grey-levels: a rigid tool cutting a deformable body in two pieces.

Discussion

We can point out some parallels and differences between smoothed and standard particle systems:

- Cohesion/pressure forces depicted in Figure 3 are very similar to Lennard-Jones forces, with short range repulsion and long range attraction. However, in contrast with Lennard-Jones forces, which result from microscopic observations, we have derived forces from a global equation. Our method could be easily generalized to other materials since another state equation would generate another set of interaction forces.
- Viscosity, modeled as in the original SPH method, gives forces that are very close to previous ad-hoc models [MP89, DG95], where damping forces were computed from relative speeds and proximities of each pair of particles.
- While symmetric pairwise forces are present in both formalisms (which results in very similar implementations), smoothed particles ensure both stability and accuracy since they belong to Monte Carlo approaches.

In conclusion, smoothed particles are algorithmically close to previous models, but are based on a more soundly motivated mathematical basis. Important benefits of this new formalism are detailed in the next sections. Among others, it naturally defines a surface around a deformable body, and gives stability criteria that help efficiency.

4 Associating a surface to smoothed particles

Since our final goal is to create animations for Computer Graphics purposes, we have to provide a continuous representation for our discretized model of matter.

Particle systems have often been coated with implicit functions [MP89, Ton91], which seems to be the easiest way to model objects with unfixed topology. But if the implicit function is chosen arbitrarily, large undesired volume variations can be produced, and their suppression requires more computation [DG95]. This problem should not occur with a coherent definition of the surface, as particle systems are supposed to keep their volume nearly constant. Fortunately, a specific benefit of smoothed particles is to give a natural way of defining a surface.

4.1 Level Sets of Mass Density

As we already consider an object as a set of smeared-out masses, we can easily define where it lies. The density ρ actually is a continuous function that indicates where and how mass is distributed in space. Isovalues of density define implicit surfaces which are coherent with the model we use for simulation. The choice of an adequate isovalue should lead to volume preservation at no extra cost.

4.2 Coherent choice of Iso-Density

We have to find an appropriate iso-contour value which should be coherent with the underlying physical model. If the particles separate into two groups that no longer interact, the isosurface should exhibit two connected components. On the other hand, a single surface should be generated by a set of interacting particles.

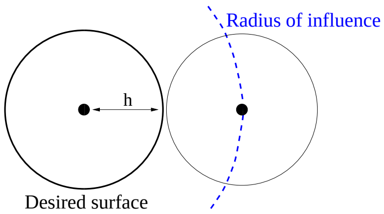

Figure 5: Two particles at their maximum interaction range.

Figure 5 shows that if two particles are a distant of $2h$ apart, no more interaction force occurs. Then, the surface should be located at a distance h from each boundary particle. In the specific case of two particles, the isovalue should be $2 \cdot W_h(h)$. Even if this choice is not compelling for several particles, we made our first experiments with that iso-value of density to display the surface. An example of animation using this technique is shown on Figure 6.

First tests with surface visualization in 2D show that the choice of our isovalue appears relatively good (see figure 7): the surface area undergoes variations of maximum ten percent. But it actually seems that the contour both preserving its surface area (i.e., its volume in 3D) better, and resulting in smooth and realistic shapes remains to be found. The relevant isosurface should surely be based on both W_h and ρ_0.

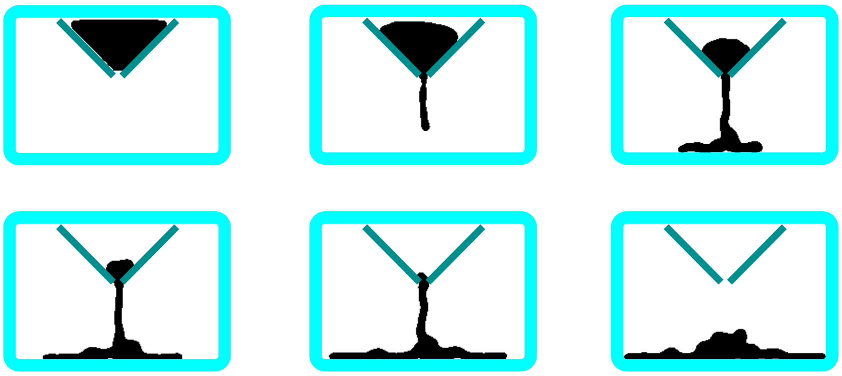

Figure 6: Snapshots from a sequence with a surface coating the particles: a quasi-liquid material (80 particles) falling under gravity.

5 Implementation issues

A basic implementation of smoothed particles would run in $O(n^2)$ with fixed time steps. This is not satisfactory since particle system simulations can require a large number of particles. Moreover, previous approaches were limited by the choice of a very short time step to avoid divergences or oscillations. This section explains how we can simulate smoothed particles in linear time thanks to an adequate data structure, and how time steps used for integration can be adapted according to a local stability criterion—a notion easier to define for smoothed particles than for conventional particle systems.

5.1 Neighbor search Acceleration

As in any N-body problem, one of the bottlenecks of the computation is the amount of time needed to perform force evaluation. In our case, forces are applied only to nearby particles, so a nearest neighbor search must be performed first to identify interacting neighbors. This search will also be used to find particles contributing to the implicit surface at a given point. So a lazy evaluation in $O(n^2)$ is not satisfactory.

As the interaction distance of each particle stays constant, the optimal data structure to perform nearest neighbor searching is a grid of voxels [MP89] of size $2h$. With such a structure, and since our particles are assumed to stay well distributed, the evaluation of forces on particles becomes $O(n)$. Creating the grid of voxels and finding particles lying in each voxel takes $O(n)$ operations too. So the whole simulation results in a linear total complexity, enabling a large number of particles.

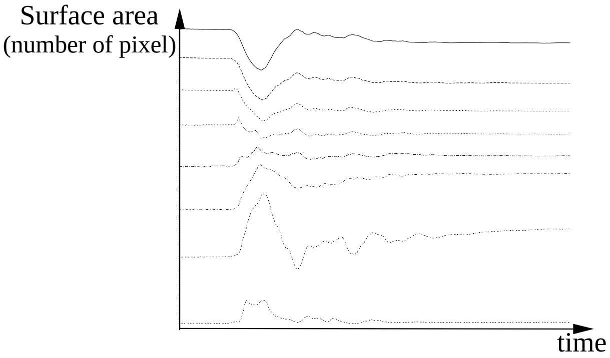

Figure 7: Volume variations for different iso-densities during the animation of Fig. 4.

5.2 Locally adaptive integration

Choosing a time step that both avoids divergence and ensures efficiency is the most important practical problem of conventional particle systems. Moreover, using a same integration time step for all the particles of a scene is not optimal since we are always tightly constrained by the shortest time scale occurring. Fortunately, local stability criteria have been naturally found for smoothed particles that in turn greatly reduce the computation. We actually perform adapted integration time steps for each particle both to reduce computation in stable areas and to automatically avoid divergence.

Time Stepping

In finite differencing methods, a stability criterion that often occurs is the Courant-Friedrichs-Lewy criterion, called here *Courant condition* [PTVF92]. It can be written as: $v\delta t/\delta x \leq 1$ where δt is the time step used for integration, v a velocity, and δx is the grid size. It intuitively means that if a phenomenon propagates with a maximum velocity v, it must not be integrated with a too large time step, or some grid points will be leaped. These leaps will result in a lack of information in these points and will give rise to instability during integration.

Translated into smoothed particles, the Courant condition requires that each particle i must not be passed by, giving a modified condition such as: $\delta t_i \leq h/c$, where h the smoothing length, and c the sound speed in the object, which is the maximum velocity of a deformation wave inside the material. The use of viscosity (see section 2.3) changes the criterion a little [HK92], giving a required time step satisfying:

$$\delta t_i \leq \delta t_i^{Courant} = \alpha \frac{h}{c + 0.6(c + 2 \cdot \max_j \ \mu_{ij})} \tag{8}$$

where α is the Courant number (approx. 0.3). To this criterion one can add others. In our implementation, we increase accuracy by taking into account the current speed of the particle, which results in an overall time step chosen so that:

$$\delta t_i = \min(\delta t_i^{Courant}, \alpha \frac{h}{v_i}).$$

Adaptive Time Integration

Now that we have defined a stability criterion, we could perform integration with a global adapted time step equal to $\delta t = \min_i \; \delta t_i$. Nevertheless, only a few particles (those undergoing large external forces) may need a precise integration while the others are quite stable. The solution is then to use individual particle time steps [HK92].

Assume a user-defined simulation rate Δt. To facilitate synchronization of integration, we only use power of two subdivisions of Δt. It means that the effective time step is chosen by finding the smallest positive n_i satisfying $\Delta t / 2^{n_i} \leq \delta t_i$. The particle positions are then advanced at every smallest time step, while force evaluations are performed at each individual time step. This method permits forces to be evaluated only when needed, offering substantial gains in performance though guaranteeing stability.

The integration scheme we use is a standard "leapfrog" integrator, which is accurate to second-order:

$$\mathbf{r}_i^{n+1/2} = \mathbf{r}_i^{n-1/2} + \mathbf{v}_i^n \delta t \tag{9}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \mathbf{a}_i^{n+1/2} \delta t \tag{10}$$

where the indices in superscripts denote at which time step the quantities are computed. Since we use velocity through viscosity to estimate acceleration, second-order accuracy is preserved if we use a predicted estimate $\tilde{\mathbf{v}}_i^{n+1/2}$ obtained from:

$$\tilde{\mathbf{v}}_i^{n+1/2} = \mathbf{v}_i^n + \frac{\delta t}{2} \mathbf{a}_i^{n-1/2}.$$

This estimate is then used to compute the time-centered acceleration $\mathbf{a}_i^{n+1/2}$, from which the new velocity $\mathbf{v}_i^{n+1}$ will be computed from equation (10).

The change of time step during integration can also induce a loss of second order accuracy if no care is taken. That is why, at each change of a particle's own time step, the particle positions must be corrected using:

$$\mathbf{r}_i = \mathbf{r}_{i,old} + \frac{(\delta t_{new}^2 - \delta t_{old}^2)}{8} \mathbf{a}_i.$$

The resulting integration algorithm performs adaptive and individual integrations of our smoothed particles. Note that this algorithm can be used as soon as a stability criterion is known. Its advantage is that the time step is no longer a user-defined parameter but is totally managed by physical and numerical stability criterions.

6 Conclusion

We have defined a new type of particle system. Instead of considering particles as point masses discretizing an object, we prefer to define smoothed particles as samples of mass smeared out in space. While modeling the object in a smoother way, the method also provides equations of motion similar to conventional particles (symmetric central forces between pairs) and offers many new opportunities. Time integration for instance can be handled in a powerful way since each particle is integrated at individual time steps. The choice of the current time step is selected according to different criteria to ensure stability. Others advantages of this technique include an implicit representation coherent with the physical model as derived from the spatial density, efficient complexity, and intuitive parameters provided for the user to choose the kind of viscous material he wants to animate. The Monte-Carlo interpretation also ensures that stability increases with particles number.

All these advantages make this method a coherent and robust way to use particle systems. But a lot of avenues are yet to be explored: varying the smoothing length will provide a spatially adaptive system, which can in turn add efficiency since integration steps are proportional to this length. We can think for instance of creating "large" particles inside objects, and "smaller" particles in the periphery, reproducing the derm-epiderm approach of [LJR+91]. Thermal conduction [Ton91] can also be taken into account with this paradigm.

Animations can be found at:
`http://w3imagis.imag.fr/~Mathieu.Desbrun/smoothed.html`

7 Aknowledgements

The authors would like to thank Eugene Fiume for pointing us to SPH and for early discussions, Andy Hanson for his continuous help, and François Faure and Rémi Cozot for a full re-reading of the paper.

References

[DG95] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. *Computer Graphics*, pages 287–290, August 1995. Proceedings of SIGGRAPH'95 (Los Angeles, CA).

[Gas93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, pages 313–320, August 1993. Proceedings of SIGGRAPH'93 (Anaheim, CA).

[GLG95] M.N. Gamito, P.F. Lopes, and M.R. Gomes. Two-dimensionnal simulation of gaseous phenomena using vortex particles. In *6th Eurographics Workshop on Animation and Simulation*, Maastricht, Netherlands, September 1995.

[GMTT89] Jean-Paul Gourret, Nadia Magnenat Thalmann, and Daniel Thalmann. Simulation of object and human skin deformations in a grasping task. *Computer Graphics*, 23(3):21–29, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).

[HK92] Lars Hernquist and Neal Katz. Treesph: A unification of sph with the hierarchical tree method. *App. J. Supp.*, 70:419, 1992.

[LHVD95] A. Luciani, A. Habibi, A. Vapillon, and Y. Duroc. A physical model of turbulent fluids. In *6th Eurographics Workshop on Animation and Simulation*, Maastricht, Netherlands, September 1995.

[LJR+91] Annie Luciani, Stéphane Jimenez, Olivier Raoult, Claude Cadoz, and Jean-Loup Florens. A unified view of multitude behaviour, flexibility, plasticity, and fractures: balls, bubbles and agglomerates. In *IFIP WG 5.10 Working Conference*, Tokyo, Japan, April 1991.

[LP95] Jean-Christophe Lombardo and Claude Puech. Oriented particles: A tool for shape memory objects modelling. In *Graphics Interface'95*, Quebec, Canada, May 1995.

[Mon92] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.*, 30:543, 1992.

[MP89] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *SIGGRAPH '89 Courses 30 notes*, pages 305–309, August 89.

[PTVF92] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C, second edition*. Cambridge University Press, New York, USA, 1992.

[PW89] Alex Pentland and John Williams. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics*, 23(3):215–222, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).

[Ree83] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *Computer Graphics*, 17(3):359–376, 1983.

[Ton91] David Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface'91*, pages 255–262, Calgary, AL, June 1991.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California).

[TPF89] Demetri Terzopoulos, John Platt, and Kurt Fleisher. Heating and melting deformable models (from goop to glop). In *Graphics Interface'89*, pages 219–226, London, Ontario, June 1989.

A Mathematical bases

Suppose we have a field f defined over all space. We can calculate a mean value within a spatial interval with:

$$< f(\mathbf{r}) > = \int_{space} f(\mathbf{r}')\, W_h(\mathbf{r}-\mathbf{r}')dr' \tag{11}$$

where $W_h(\mathbf{r})$ is a smoothing kernel, and h the smoothing length specifying the extent of the averaging volume. $W_h(\mathbf{r})$ is peaked about $\mathbf{r}=0$ so that it tends to the Dirac delta function as $h \to 0$, keeping its integral normalized to 1. Expanding the integrand of (11) as a power series in h gives, if $W_h(\mathbf{r})$ is an even function:

$$< f(\mathbf{r}) > = f(\mathbf{r}) + O(h^2)$$

Suppose now that our kernel W_h has a finite support. Then we can approximate any derivative of f easily. For instance, we have by definition in 1D:

$$< \frac{\partial f}{\partial x}(r) > = \int_{-\infty}^{+\infty} \frac{\partial f}{\partial x}(r')\; W_h(r-r')dr'.$$

Integrating by parts, this yields:

$$\int_{-\infty}^{+\infty} \frac{\partial f}{\partial x}(r')\; W_h(r-r')dr' \;= [f(r')\; W_h(r-r')]_{r=-\infty}^{r=+\infty} - \int_{-\infty}^{+\infty} f(r')\cdot(-1)\cdot\frac{\partial W_h}{\partial x}(r-r')dr',$$

so that we can write:

$$< \frac{\partial f}{\partial x}(r) > = \int_{-\infty}^{+\infty} f(r')\; \frac{\partial W_h}{\partial x}(r-r')dr'. \tag{12}$$

It proves that there is no need to know analytical derivatives to calculate their mean values.

B Pressure forces

A natural way to write the pressure gradient in SPH formalism is:

$$\nabla P(\mathbf{r}) \;=\; \sum_j m_j \frac{P_j}{\rho_j} \nabla W_h(\mathbf{r}-\mathbf{r}_j).$$

We can use it in the equation of motion when pressure is exerted on a small volume dV of mass m:

$$m\,\mathbf{a} = \rho\, dV\, \mathbf{a} = \mathbf{F} = -\nabla P\, dV,$$

where $\mathbf{a}$ is the acceleration and $\mathbf{F}$ the pressure force on dV. But a quick observation shows that the action-reaction principle would then not be enforced since $m_i \nabla P_i/\rho_i$ is not equal to $m_j \nabla P_j/\rho_j$ for $i \neq j$. To symmetrize the pressure forces, we can use the simple derivation rule $\nabla P/\rho = \nabla(P/\rho) + P\nabla\rho/\rho^2$ to write:

$$\frac{\nabla P_i}{\rho_i} = \sum_{j\neq i} m_j \frac{P_j}{\rho_j^2} \nabla_i W_h^{ij} + \frac{P_i}{\rho_i^2} \sum_{j\neq i} m_j \nabla_i W_h^{ij}$$

where the notation $\nabla_i W_h^{ij}$ denotes the gradient of $W_h(\mathbf{r}_i - \mathbf{r}_j)$ taken with respect to the coordinates of particle i. The equation of motion for a particle is then:

$$\mathbf{a}_i = \frac{\mathbf{F}_i}{m_i} = -\sum_{j\neq i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) \nabla_i W_h^{ij}.$$