SPH survival kit

Although SPH is an excellent method in many cases, it is fairly hard to find a set of parameters that give stable and realistic simulations, and it can also be difficult to find parameters that rescale a simulation (e.g. from 0.1 l of water to 10 l of water).

Here, we propose some improvements that will give more stable simulations. However, note that it is indeed possible to get stable simulations with the previously given parameters, but the timestep must be held perhaps too small for the simulation to be interactive.

Parameters

Although one would like the SPH parameters to have direct correspondence to the real world, this is in general not the case. The main problem when simulating water is that the speed of sound is usually set to a value much smaller than the real value (e.g. 1 m/s instead of 1500 m/s). This results in a compressible system, and in turn this results in water with a too high density.

Therefore, for a given sound velocity and timestep, try to lower the mass per particle so that the computed average density becomes closer to 1000 kg/m3, i.e. so that the pressure, $p = c_2 (\rho - \rho_0)$ fluctuates around zero.
This will also affect the resulting volume, and thus the density has a non-linear dependence on the particle mass, but by testing a couple of times you should at least find a much better value.

Just to complicate this a bit ☺
If the density is lowered and particles are more sparse, one might need to use a larger smoothing length. This will also affect the computed density, since, in particular, the self contribution from the center particle depends a lot on the smoothing length (short smoothing length = large self contribution, and there is also a lower bound, i.e. the self contribution to the density should not be larger than ρ_0).

Different ways of computing field differentials

When computing forces/acceleration in SPH, Newton’s third law tells us that in order to conserve linear momentum (“rörelsemängd”), pairwise forces must be equal in size with opposite sign. Therefore the differentials in Navier Stokes equations that give rise to these forces, must also be symmetrized. This can be done in several ways, leading to different types of averages, and in the end it can affect the simulation, in particular in extreme situations, e.g. near a boundary, or at very short distances.

Referring to the SPH lecture notes, the gradient in the SPH pressure force can thus take at least the two following different forms (in fact, there is a very recent article looking into this![1]):

$$P_{ij} = -m_j \left[\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right],$$

as previously discussed in the lecture notes, or,

$$P_{ij} = -m_j \left[\frac{p_i + p_j}{2\rho_i \rho_j} \right].$$

According to Ref [1], this *latter* expression gives more accurate differentials, and this is also my own experience, in particular near boundaries where densities can be small due to lack of neighbours on the boundary side.

Kernel functions

Kernel functions smoothes out the field on each particle by averaging it over several particles. The averaging is weighted by the kernel function and therefore the choice of kernel function will have some effect on the simulation. Kernel functions most fulfil a number of requirements, e.g. having limited range (“compact support”), being normalized, being differentiable, etc. and there is a wide choice of functions satisfying this. It is also possible to use different kernels depending on what is computed, i.e. one for the pressure gradient, one for the density, one for the viscosity.

As an alternative to just using poly6 for everything, the following choice is know to have some advantages.

For the density use poly6, i.e.

$$K(r) = \frac{315}{64\pi h^9}\left(h^2 - r^2\right)^3 \qquad \text{(zero for } r > h\text{)}$$

For the pressure gradient, use a “spiky kernel”, or rather its gradient,

$$K(r) = \frac{15}{\pi h^6}\left(h - r\right)^3$$

(zero for $r > h$)

$$\text{grad}(K(r)) = -\frac{45}{\pi h^6}\left(h - r\right)^2 \hat{\mathbf{r}}$$

For the viscosity use another polynomial kernel, or rather its Laplacian,

$$K(r) = -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1$$

(zero for $r > h$)

$$\text{Laplacian}(K(r)) = \frac{45}{\pi h^6}(h - r)$$

These choices give more repulsive pressure at short distances, and thus avoids clustering. It also gives a more stable viscosity, and this in turn makes it possible to damp the simulation better, which often is required when doing interactive simulations with fairly large timesteps. Note that the above expressions assume that $r_{ij} = r_i – r_j$. With $r_{ij} = r_j – r_i$instead, there will be a sign change in the gradient expression due to the inner derivative.

/Kenneth

References

1. F. Colin, R. Egli and F.Y. Lin, Computing a null divergence velocity field using smoothed particle hydrodynamics, Journal of Computational Physics, In Press, Corrected Proof, , Available online 24 February 2006. http://www.sciencedirect.com/science/article/B6WHY-4JBGKMW-5/2/0643112f85fe9f313e86911b63a6d890)