CS205b/CME306

Lecture 3

1 Smoothed Particle Hydrodynamics (SPH)

1.1 Representation and Simulation

In the previous lecture, we showed how to define density throughout space by smearing out masses at discrete points in space.

$$\rho(x) = \sum_i m_i W(x - x_i)$$

We did not describe how one might choose those masses m_i or the locations x_i where they live. If we instead have some initial conditions, we would like to choose a suitable set of m_i and x_i so that the density profile approximates the initial conditions. Given a certain number k of particles, we can choose k values m_i and k values (in 1D) x_i giving us $2k$ variables. We can write down $2k$ equations by making $2k$ measurements of ρ, and solve for a suitable set of initial masses and locations, though run into issues with overdetermined/underdetermined systems.

This gives us some connection between the (piecewise) continuous representation and a discrete representation. The idea of numerical simulation is to evolve the discrete version forward in time. The continuous one is what we see in the real world, and it evolves forward in time based on some physical laws. If at some later time we could compare the continuous one to the discrete one, we would like to have the two match in some way. There will be errors between the two, which arise from two sources: errors in the initial discretization and errors that accumulate during the course of the simulation. In this way, we may use numerical simulation to predict the outcome of a continuous system.

1.2 Other Attributes

We are not, however, limited to attaching density to our chunks. We can attach any attribute A_i to the chunks and use $W(x)$ to distribute it. This is typically done using

$$A(x) = \sum_i \frac{m_i}{\rho_i} A_i W(x - x_i)$$

where $\rho_i = \rho(x_i)$. Note that we get back our definition of $\rho(x)$ if we let $A_i = \rho_i$. The extra scaling factor is a volume weighting that cancels out the weighting in $W(x)$. If we integrate the contribution of one chunk throughout space,

$$\int_{-\infty}^{\infty} A(x)\, dx = \int_{-\infty}^{\infty} \frac{m_i}{\rho_i} A_i W(x - x_i)\, dx = \frac{m_i}{\rho_i} A_i$$

which is just the volume weighted attribute as one would expect.

Because we have a smooth definition of $A(x)$ everywhere, we may compute its derivatives. For example, ∇A can be computed as

$$\nabla A = \sum_i \frac{m_i}{\rho_i} A_i \nabla W(x - x_i).$$

We have now described how to represent mass, other scalar quantities, and derivatives of these scalar quantities in space just based on the idea of having quantities attached to attributes and moving these attributes around. We automatically conserve mass with this method, so the next step is to consider momentum.

2 Conservation of Momentum (Forces)

At this point we would like to actually simulate something. We have not yet introduced any means by which this motion can be computed, and this is where forces are introduced.

Newton's second law provides the necessary relationship between forces and motion and may be written as $F = ma = p'$, where p is the momentum of a particle. One may also view this relationship as an extension to conservation of momentum. Note that Newton's first law is conservation of momentum $\sum_i p_i =$ constant, which a consequence of $\sum_i F_i = 0$, a system experiencing no external forces. Newton's third law (equal and opposite reactions) requires that forces between particles occur in equal and opposite pairs, so in the absence of external forces, net force is still zero and the momentum of the system is conserved. Newton's third law provides a convenient means for enforcing conservation of momentum in the Lagrangian framework (we will use this when formulating springs later).

Particles may now be evolved through space as long as the forces acting on them can be computed. One of the simplest and most important forces is gravity. For our purposes, gravity may assumed to be constant throughout space. Then, we compute the force on a particle due to gravity as $F = mg$, so that a particle experiencing no other forces simply falls with constant acceleration $a = g$. A somewhat more interesting force is a simple drag force $F = -kv$, where k is constant and v is the particle's velocity. A particle with higher velocity feels more drag, and the resulting force opposes its motion. A particle experiencing only this force slows down but never reaches rest or changes its direction.

3 Linearized System

Force is in general a function of both the positions and velocities of the particles in a system. That is, $F = F(x, v)$. In the interests of writing down a linear system to analyze stability, it is convenient to approximate this force as $F(x, v) \approx F(x_0, v_0) + F_x(x, v)(x - x_0) + F_v(x, v)(v - v_0)$ and also ignore the inhomogeneous term $F(x_0, v_0)$, since Duhamel's principle states that the inhomogeneous term does not affect stability. Note that this approximation omits gravity, since it is inhomogeneous. We can typically make these simplifications when looking at stability. Forces of the form $F_x(x, v)x$ are rather like spring forces, and forces of the form $F_v(x, v)v$ behave like damping forces. Using $F_x = ma_x$ and $F_v = ma_v$, the motion of the particle is described by the first order linear system

$$\begin{pmatrix} x \\ v \end{pmatrix}' = \begin{pmatrix} 0 & 1 \\ a_x & a_v \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix}.$$

The eigenvalues of this system are

$$\lambda = \frac{a_v \pm \sqrt{a_v^2 + 4a_x}}{2}$$

and have units of Hertz (s^{-1}). Solutions look $e^{\lambda t}$, so well-posedness requires the real part of the eigenvalues be nonpositive, $\mathrm{Re}(\lambda) \leq 0$. This places some restrictions on the way we can model forces of nature to prevent the system from blowing up.

It is necessary for $a_x \leq 0$. If a particle experiences no force at the origin but experiences a stronger force as it moves away, that force should be a restoring force rather than one that push it away harder as it moves farther away and causes exponential growth.

Similarly, it is necessary for $a_v \leq 0$. A force that did not satisfy this would tend to apply forces in the direction of motion that get stronger as the particle moves faster and result in exponential growth.

When $-a_v < 2\sqrt{-a_x}$, we call the system under-damped. The eigenvalues contain imaginary components, and the solution exhibits period behavior. If $a_v < 0$, the system has exponential damping. If $a_v = 0$, the system is undamped. Note that in the undamped case, the eigenvalues are pure imaginary. In the under-damped case $|\lambda| = \sqrt{-a_x}$ does not depend on the amount of damping applied.

When $-a_v > 2\sqrt{-a_x}$, we call the system over-damped. The eigenvalues are real and distinct, and the solution exhibits exponential decay only.

When $-a_v = 2\sqrt{-a_x}$, we call the system critically damped. The eigenvalues are equal, and the solution exhibits exponential decay with at most one overshoot. A critically damped system decays faster than it would have with any other amount of damping. If it is under-damped, it overshoots repeatedly and decays slowly. If it is over-damped, the excessive damping causes it to move towards equilibrium slowly. Severe damping effectively freezes the system so it can hardly move.