CS667 **Lecture 14: Particle System Dynamics** 17 March 2005

Jeff Wang Lecturer: Steve Marschner

1 Introduction

In computer graphics, dynamic simulations can become extremely complicated and can vary from situation to situation. Solution methods should not appply to just one specific situation, but flexible enough so that they can be applied to a range of problems.

2 Particle Systems

Particle systems are a common way to simulate the laws of dynamics. They consist of N number of particles, each with a mass, mi, concentrated at one point as well as their own state variables. Examples of these state variables are position $\overrightarrow{x_i}$ and momentum, where p = mv.

The equations of motion for particle systems are:

$$\frac{d}{dt}\overrightarrow{x_i} = \frac{1}{m}\overrightarrow{p_i} \tag{1}$$

$$\frac{d}{dt}\overrightarrow{p_i} = \overrightarrow{F_i} \tag{2}$$

where $\overrightarrow{F_i}$ is the net force on particle i. Note that we could have just written this as

$$\frac{d^2}{dt^2}\overrightarrow{x_i} = \frac{1}{m}\overrightarrow{F_i} \tag{3}$$

but this results in a rather ugly 2nd order equation.

Examples of practical uses for particle systems include:

1. Fireworks - see figure 1.

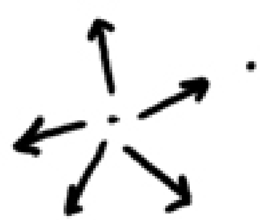

Figure 1: Each firework shell is a particle and is subject to gravity and aerodynamic drag.

2. Double Pendulum - see figure 2.
3. Cloth - see figure 3.

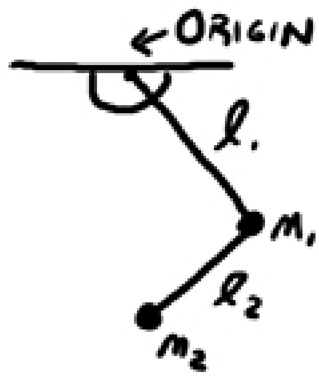

Figure 2: Two points masses are connected by two infinitely stiff bars. The system is free to revolve about the origin.

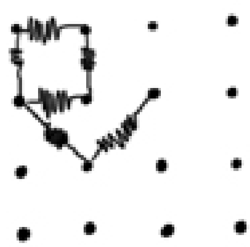

Figure 3: Cloth is estimated as a grid of particles connected by springs. Some particles have springs that connect diagonally to prevent bending.

3 Forces

There are essentially two types of forces that we are interested in:

1. External forces $\overrightarrow{F_i^{(e)}}$
   Examples: gravity, immovable objects, user inputs
2. Internal forces $\overrightarrow{F_{ji}}$, or from interactions between particles
   Examples: springs, repulsion forces for collision
   The above notation, $\overrightarrow{F_{ji}}$, refers to an internal force on i due to j. Remember, that according to Newton's Third Law, $\overrightarrow{F_{ji}}$ = -$\overrightarrow{F_{ij}}$

Thus the total force on particle i is:

$$\overrightarrow{F_i} = \sum_j \overrightarrow{F_{ji}} + \overrightarrow{F_i^{(e)}} \tag{4}$$

However, we must put this equation into a more general form, using matrices, for ODE solvers. The matrix describing the state of particle i has 6 variables:

$$\overrightarrow{Y_i} = \begin{bmatrix} \overrightarrow{x_i} \\ \overrightarrow{p_i} \end{bmatrix} \tag{5}$$

The matrix describing state of all particles $\mathcal{Y}$, has $6N$ variables:

$$\mathcal{Y} = \begin{bmatrix} \overrightarrow{Y_i} \\ \vdots \\ \overrightarrow{Y_N} \end{bmatrix} \tag{6}$$

So, in standard form, to describe the change in the system over time, we can write:

$$\tfrac{d}{dt}\mathcal{Y} = f(\mathcal{Y})$$

where f is some function of state. But this equation is incomplete in that it does not consider what causes the change in the system (forces). We introduce a level of abstraction for forces into the equation by rewriting it as:

$$\frac{d}{dt}\mathcal{Y} = f(\mathcal{Y}, \mathcal{F}(\mathcal{Y})) \tag{7}$$

where $\mathcal{F}$, is a matrix of $3N$ variables and is defined as $\mathcal{F} = \begin{bmatrix} \overrightarrow{F_i} \\ \vdots \\ \overrightarrow{F_N} \end{bmatrix}$

Then, we can re-write f so that it is defined per particle:

$$f_i(\mathcal{Y}, \mathcal{F}) = f_i(\overrightarrow{Y_i}, \overrightarrow{F_i}) = \begin{bmatrix} \frac{1}{m_i}\overrightarrow{p_i} \\ \overrightarrow{F_i}(\mathcal{Y}) \end{bmatrix} \tag{8}$$

Since $\overrightarrow{F_i}(\mathcal{Y})$ has the $\mathcal{Y}$ term, it is dependent on what all of the particles are doing. Thus, we have to write a function for the sum of all forces experienced by each particle in the system.

3.1 Simple example: Particles under gravity and air resistance

Using the equations we've developed, a simulation can be written for the particles in a fireworks display. As previously mentioned, our first step is to describe the forces experienced by each particle in the system. We write the following equation that describes the forces, based on the assumption that the z-axis is up:

$$\overrightarrow{F_i^{(e)}} = \begin{bmatrix} 0 \\ 0 \\ -m_i g \end{bmatrix} + -k_1 \tfrac{\overrightarrow{p_i}}{m_i} + -k_2 \tfrac{|\overrightarrow{p_i}|\overrightarrow{p_i}}{m_i^2}$$

The first right-hand side term is obviously gravity. The following terms are due to viscous drag and aerodynamic drag, respectively. We can also make simplifications to this equation. First we can re-write velocity as a scalar term, $\frac{|\overrightarrow{p_i}|}{m_i}$, so that we are just left with the terms for motion as a scalar times a unit direction:

$$\overrightarrow{F_i^{(e)}} = \begin{bmatrix} 0 \\ 0 \\ -m_i g \end{bmatrix} + -k_1 \left(\frac{|\overrightarrow{p_i}|}{m_i}\right)^2 \frac{\overrightarrow{p_i}}{|\overrightarrow{p_i}|} + -k_2 \left(\frac{|\overrightarrow{p_i}|}{m_i}\right) \frac{\overrightarrow{p_i}}{m_i^2}$$

Finally, for this example, $\overrightarrow{F_{ji}} = 0$ as the particles are assumed to not interact with each other.

3.2 Another example: Back to the double pendulum

We are going to approximate the infinitely rigid rods as very stiff springs. To describe the forces in the system, we write:

$$\overrightarrow{F_1^{(e)}} = -k(|\overrightarrow{x_1}| - l_1)\frac{\overrightarrow{x_1}}{|\overrightarrow{x_1}|} + -c(\frac{d}{dt}|\overrightarrow{x_1}|)\frac{\overrightarrow{x_1}}{|\overrightarrow{x_1}|}$$

$$-\overrightarrow{F_{12}} = \overrightarrow{F_{21}} = -k(|\overrightarrow{x_2} - \overrightarrow{x_1}| - l_2)\frac{x_2 - x_1}{|x_2 - x_1|} - c(\frac{d}{dt}|\overrightarrow{x_2} - \overrightarrow{x_1}| - l_2)\frac{x_2 - x_1}{|x_2 - x_1|}$$

As you can see, the springs were added in between the two point masses. If we crank up the spring constant k, this would yield a better approximation of what's happening in the system; otherwise, it might look like that the rods are changing length. However, we get a stiffer system of equations which might get stability problems. Note that c is required to damp out any oscillations that might occur.

We can apply constraints by saying:

$$g_1(\mathcal{Y}) = |\overrightarrow{x_1}| - l_1 = 0$$

$$g_2(\mathcal{Y}) = |\overrightarrow{x_2} - \overrightarrow{x_1}| - l_2 = 0$$

By giving it constraints, we let it solve for all forces.

Articulated figures with rigid bodies, such as joints, can be done in a fashion similar to this approach. Remember to express everything in terms of ODE's.

4 Rigid Body Simulation

We can describe a rigid body as a continuous distribution of mass with a density, $\rho(x, y)$. One way to think of it is as a grid of rigidly stiff springs. The key quantaties are:

- Mass, $M = \int_B \rho dA$
- Center of Gravity, $\overrightarrow{C} = \int_B \rho \overrightarrow{x} dA(\overrightarrow{x})$
- Momentum, $\overrightarrow{p} = \frac{d}{dx}\overrightarrow{C}$
  Note that $\overrightarrow{C}$ is used because the rigid body will move about its center of gravity.
- Change in Momentum, $\frac{d}{dt}\overrightarrow{p} = \sum F$
- Moment of Inertia, $I_c = \int_B \rho(\overrightarrow{x} - \overrightarrow{c})dA(\overrightarrow{x})$
  For 2D, this value is a scalar. For 3D, it's a vector.
- Angular Momentum, $L = I_c\omega$ where $\omega = \frac{d}{dt}\theta$
- Torque, $N = \overrightarrow{F} \times \overrightarrow{r}$, cause change in angular momentum
- Change in Angular Momentum, $\frac{d}{dt}L = \sum N_i$

Given these items, we re-write the matrices that describe the system state.

$$\mathcal{Y} = \begin{bmatrix} \overrightarrow{C} \\ \overrightarrow{p} \\ \theta \\ \omega \end{bmatrix} \tag{9}$$

$$\frac{d}{dt}\mathcal{Y} = \begin{bmatrix} \frac{1}{m}\overrightarrow{p} \\ \frac{1}{I}\overrightarrow{L} \\ \overrightarrow{N} \end{bmatrix} \tag{10}$$

For 3D rigid body dynamics, the equations can be somewhat scaled up by adding another dimenson. Note that the rotation matrix is now a 9 element matrix and 4-vector quaternion matrix is introduced. The moment of inertia in 3D is a tensor matrix. For more information on 3D rigid body dynamics, consult links to the SIGGRAPH course notes among others on the 667 website.