

Particle Dynamics

Reading

Particle Systems Dynamics handout

Optional:

Hockney and Eastwood. *Computer simulation using particles*. Adam Hilger, New York, 1988.

Gavin Miller. "The motion dynamics of snakes and worms."
Computer Graphics 22:169-178, 1988.

2

Overview

- One lousy particle
- Particle systems
- Forces: gravity, springs
- Implementation

3

Newtonian particle

- Differential equations: $f=ma$
- Forces depend on:
- Position, velocity, time

$$\frac{d^2x}{dt^2} = \frac{f(x, \dot{x}, t)}{m}$$

4

Second order equations

$$\ddot{x} = \frac{f(x, \dot{x}, t)}{m} \quad \text{Has 2}^{\text{nd}} \text{ derivatives}$$

$$\begin{bmatrix} \dot{x} = v \\ \dot{v} = \frac{f(x, \dot{x}, t)}{m} \end{bmatrix} \quad \text{Add a new variable } v \text{ to get a pair of coupled 1}^{\text{st}} \text{ order equations}$$

5

Phase space

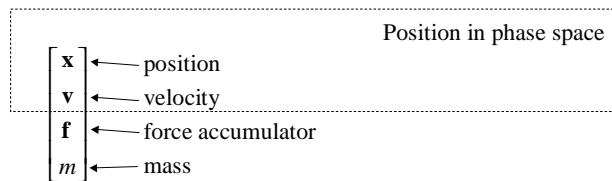
$$\begin{bmatrix} x \\ v \end{bmatrix} \quad \text{Concatenate } x \text{ and } v \text{ to make a 6-vector: position in phase space}$$

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} \quad \text{Velocity on Phase space: Another 6-vector}$$

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix} \quad \text{A vanilla 1}^{\text{st}}\text{-order differential equation}$$

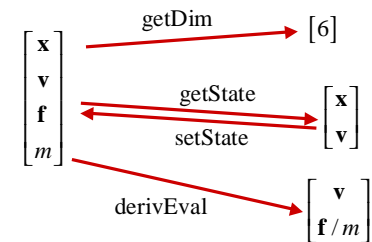
6

Particle structure



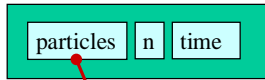
7

Solver interface



8

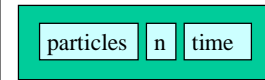
Particle systems



$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \dots \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix}$$

9

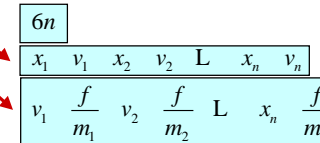
Solver interface



get/setState

getDim

derivEval



10

Differential equation solver

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f}/m \end{bmatrix}$$

Euler method: $x(t+h) = x(t) + h \cdot \dot{x}(t)$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \cdot \dot{\mathbf{x}}_i$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t \cdot \dot{\mathbf{v}}_i$$

Gets very unstable for large Δt

Higher order solvers perform better: (e.g. Runge-Kutta)

11

derivEval loop

1. Clear forces
 - Loop over particles, zero force accumulators
2. Calculate forces
 - Sum all forces into accumulators
3. Gather
 - Loop over particles, copying v and f/m into destination array

12

Forces

- Constant (gravity)
- Position/time dependent (force fields)
- Velocity-dependent (drag)
- N-ary (springs)

13

Force structures

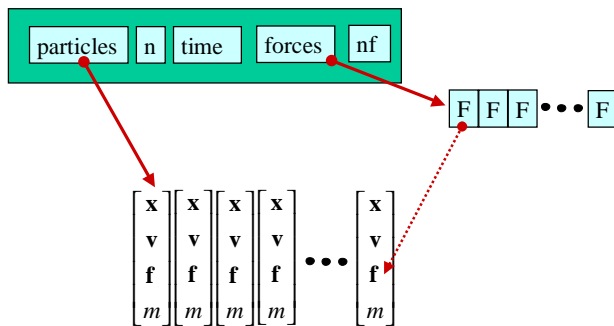
Force objects are black boxes that point to the particles they influence, and add in their contribution into the force accumulator.

Global force calculation:

- Loop, invoking force objects

14

Particle systems with forces



15

Gravity

Force law:
 $\mathbf{f}_{grav} = m\mathbf{G}$

`p->f += p->m * F->G`

16

Viscous drag

Force law:

$$\mathbf{f}_{drag} = -k_{drag} \mathbf{v}$$

`p->f -= F->k * p->v`

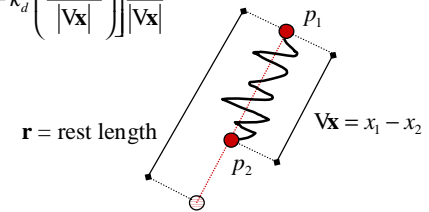
17

Damped spring

Force law:

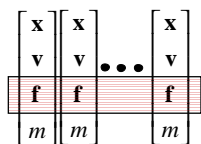
$$\mathbf{f}_1 = - \left[k_s (|\mathbf{Vx}| - r) + k_d \left(\frac{\mathbf{VvVx}}{|\mathbf{Vx}|} \right) \right] \frac{\mathbf{Vx}}{|\mathbf{Vx}|}$$

$$\mathbf{f}_2 = -\mathbf{f}_1$$



18

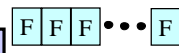
derivEval Loop



Clear force accumulators

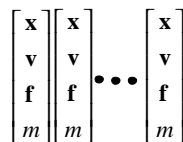
1

2



Apply forces to particles

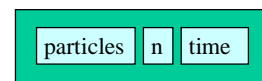
3



Return [v,f/m,...] to solver

19

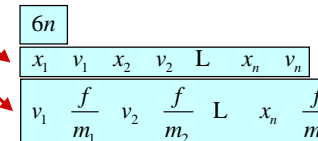
Solver interface



get/setState

getDim

derivEval



20

Differential equation solver

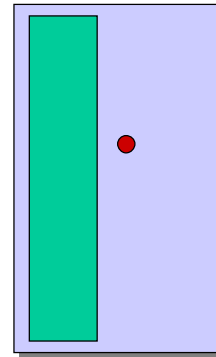
$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

Euler method:

$$\begin{bmatrix} x_1^{i+1} \\ v_1^{i+1} \\ \vdots \\ x_n^{i+1} \\ v_n^{i+1} \end{bmatrix} = \begin{bmatrix} x_1^i \\ v_1^i \\ \vdots \\ x_n^i \\ v_n^i \end{bmatrix} + \Delta t \begin{bmatrix} v_1^i \\ f_1^i/m_1 \\ \vdots \\ v_n^i \\ f_n^i/m_n \end{bmatrix}$$

21

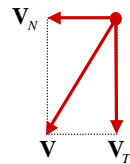
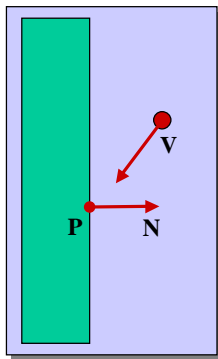
Bouncing off the walls



- Add-on for a particle simulator
- For now, just simple point-plane collisions

22

Normal and tangential components

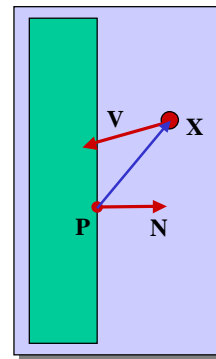


$$V_N = (N \cdot V)N$$

$$V_T = V - V_N$$

23

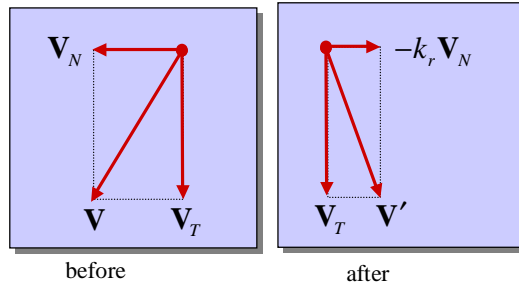
Collision Detection



- $(X - P) \cdot N < \epsilon$ Within ϵ of the wall
- $N \cdot V < 0$ Heading in

24

Collision Response



$$\mathbf{V}' = \mathbf{V}_T - k_r \mathbf{V}_N$$

25

Summary

- Physics of a particle system
- Various forces acting on a particle
- Combining particles into a particle system
- Euler method for solving differential equations

26