Particles & Shape Grammars

Kinematics

*kinematics* – motion (trajectories only – without consideration of physical considerations such as forces, friction, inertia etc.) as opposed to *dynamics*.

angle $\theta$, position $p$

\[
p = F(\theta) \quad \text{(forward) kinematic equation}
\]

\[
\theta = F^{-1}(p) \quad \text{inverse kinematic equation}
\]

We need the set of angles $\theta$ for animation.

Inverse kinematics is difficult since $F^{-1}$ may not be well-defined (there may be several or no way to achieve position $p$).

Alternative: *keyframe animation* (*inbetweening* or *interpolation*)

Newtonian Particles

position $p$, velocity $v$, state $Y := \begin{bmatrix} p \\ v \end{bmatrix} \in \mathbb{R}^6$ as functions of time $t$

\[
Y' = \begin{bmatrix} p' \\ v' \end{bmatrix} = \begin{bmatrix} v \\ \text{force/mass} \end{bmatrix} =: g(Y, t)
\]

To determine the state at time $t + h$ from the state at time $t$ need to solve the ODE.

**Examples of force $f$:** (1) *gravitational force*

\[
f = \begin{bmatrix} 0 \\ -\gamma \\ 0 \end{bmatrix}
\]

(2) *spring force* (harmonic oscillator) between $p$ and $q$.

Define $d := p - q$ and $d := \|d\|$. (Hooke's law + damping):

\[
f = -(k_{\text{spring}}(d - d_0) + k_{\text{damp}}d) \cdot \frac{d'}{dt}
\]

\[
\begin{array}{c}
\text{q} \\
\text{p}
\end{array}
\]
repulsing force

\[ f = -k_{\text{repulse}} \frac{1}{d^2} (p - q) \]

avoid \( O(n^2) \)

**Solving the ODE**

Euler (should not be used! But explains basic forward-solving idea)

\[ Y(t + h) = Y(t) + hY'(t) + O(h^2) = Y(t) + hg(Y, t) + O(h^2) \]

stiff ODE’s, Runge Kutta solvers, Verlet Integration!
Implicit Euler: \( Y(t + h) = Y(t) + hg(Y(t + h), t + h) \)

**Constraints**

*soft* (approximate) and *hard* e.g. collision

- Detection — difficult! bounding boxes, quadtrees, binary space partitioning
- Reaction: Let \( \bar{p} := p(t) - p(t^*) \) where \( t \) is the previous time and \( t^* \) the time of intersection, \( n \) is the normal at \( p(t^*) \).
  Then the new point is in the direction
  \[ p + 2(\bar{p} - (\bar{p} \cdot n)n). \]
This corresponds to an elastic collision (no energy loss, deformation).

- Soft constraints with penalty function: \( \min \| \mathbf{p} - \mathbf{p}_0 \|^2 \).

**Examples**

- `bounce.c`
- sphere bouncing: intercept with sphere of radius \( r \)

\[
\| (1 - \lambda) \mathbf{p}(t) + \lambda \mathbf{p}(t + h) \| = r
\]

Normal: \( \frac{(1 - \lambda) \mathbf{p}(t) + \lambda \mathbf{p}(t + h)}{r} \).
Language based models

*tree, shape grammar*

Example (Koch curve, snowflake) $F \rightarrow FLLRFFL$  

![Example Diagram](image)

There exists a unit circle so that the Koch curve inside it has infinite length!

Example $F \rightarrow F[R[F][F][L][F]]F + \text{random}$

OpenGL/LRfrac

Space filling Hilbert curve

![Hilbert Curve Diagram](image)

Fractals

self-similar structures
the fractal dimension \( \text{dim} \): pieces \( \cdot \) length\( ^{\text{dim}} = 1 \).

\[
\text{dim} = \frac{\ln (\text{new pieces})}{\ln (\text{old edge length})}
\]

\( \ln(20)/\ln(3) = 2.726833.. \)

**Fractal mountains**

brownian (random) “motion” (displacement) – underlying Gaussian with variance: \((\text{length of old curve})^{2(2-\text{dim})}\) yields fractal dimension \( \text{dim} \).

\[
q \quad p
\]
\[
\frac{p+q}{2} \]

(a) (b)

**Mandelbrot set**

(see also: Julia set)

\[ z_{k+1} = z_k^2 + c, \quad z_0 = 0, \quad z = x + iy \in C \]

\text{Cex/Angel/mandelbrotcrash}

Mandelbrot set == the set of all \( c \) for which this iteration converges.