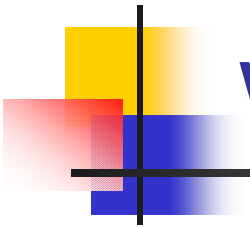
Physics for Virtual Reality

Kenneth Holmlund

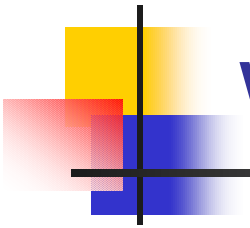
HPC2N



Why Physics?

Because most things in our everyday environment can be described by physics - and a common ambition in VR is to describe an environment.

Physics works!



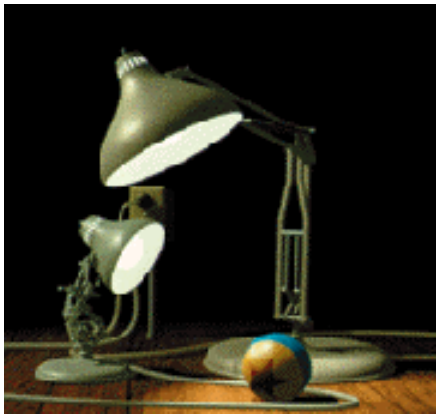
Why Physics?

Physical simulations capture complexity.

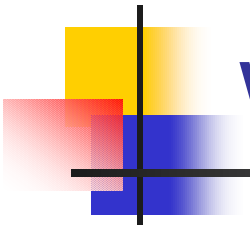
Complex behaviour *emerges* from simulation.

Why Physics?

- Humans are onthological physicists and therefore physics is important for experience, senses, emotions, humour, action, ...



- *Luxo Jr* by John Lasseter, Pixar 1987
  Keyframed realistic animation. No physical simulation. Still, physics does the trick...
- High level motion control: Jump from A to B
- Luxo Jr. made people start thinking about physics...



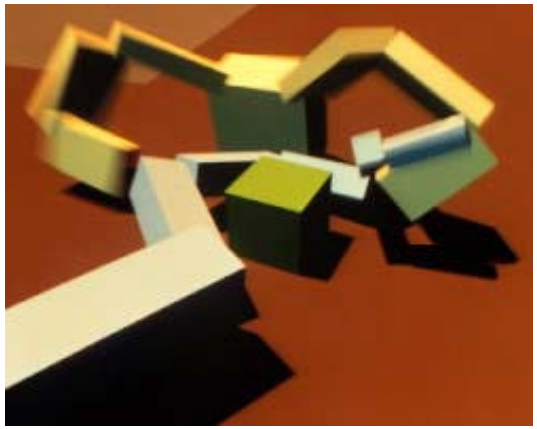
Why Physics?

Physics is better…

- Systematic
- Scalable
- Consequent
- Controllable
- Extensible
- General - does not depend on context. Therefore library software and expertize can evolve.

Why Physics?

Because we think VR and Physics in combination may open new doors to knowledge about nature.



VR is an important catalyst for many other fields!

Themes

- Dynamics (Newtons Laws)
- Rigid Body Dynamics, Interaction, Collisions
- Mechanics of Materials
- Fluids, water (water, yacht)
- Gases, smoke, clouds (smoke, train, interactive, clouds)
- Plasmas, fire, lightning, sparks (candle)
- Particles (often for special effects, explosions, fountain, hair, fur,...)
- Fields

- Many body interaction
- Collective phenomena, complex systems, emergence, ...

Themes

Also...

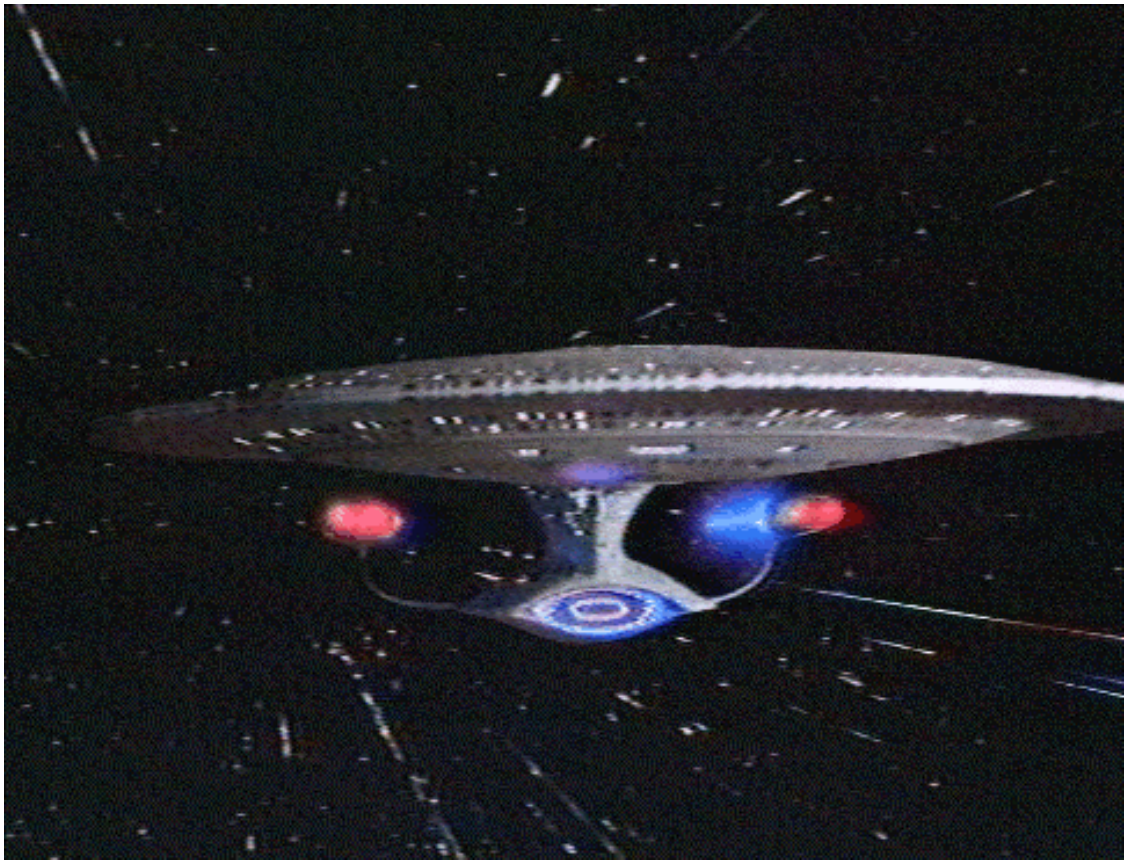
- Audio – wave tracing, acoustics, damping, ...
- Light – optics, ray tracing, ...
- Effectors and sensors – display and interaction, e.g. haptics

- The computer, the network, the, the ...
  everything – of course...

Plausibility

Why always be realistic?

- Enterprise turns!
- ”Internal inertia dampers”
- Game play



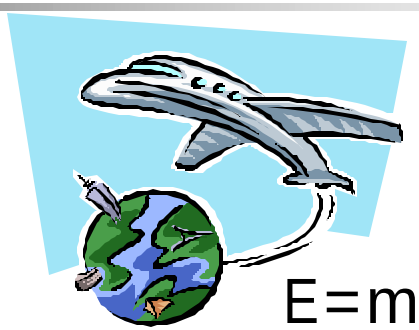
- No problem! We can mathematically modify the laws of nature. Physics and mathematical models still give us much better control than sloppy ad-hoc models!

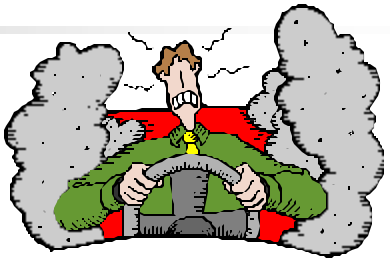
Some more examples

- Vehicles, e.g., Harvester (1, 2, 3, 4)
- Walking, running, jumping
- Collapsing structures, fracture (1)

Five easy steps...

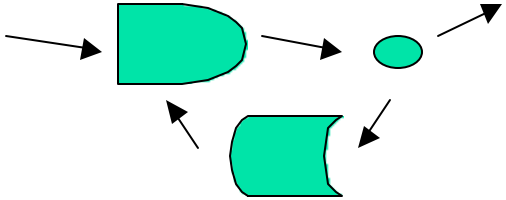
1. Physical process





2. Model – equations

$E=mc^2$



3. Simulation algorithm

4. Computer program

```
main()
...
```

5. Simulate!

Classification and definitions

- Dynamical system
  A system of ”objects” that moves or changes state due to forces, torques or possibly other causes.

- Kinematical system
  A system of ”objects” that move, but where we have no knowledge about why.

- Simulation
  Simulation of a mathematical model of a dynamical system on a computer.

Classification and definitions

- Off-line simulation
  Batch job, start and waaaait for the result, visualize, change the parameters, start and waaaaait, etc, ...

- On-line simulation
  Now we're getting somewhere – more like the real thing. "Rapid prototyping".

- Interactive simulation
  Human-in-the-loop, but no guarantee for real-time dynamics (not always a problem, e.g. for colliding galaxies or protein-DNA interaction on the pico-second scale)

- Real-time simulation
  Yes! Here, a simulation second equals a real second!

Classification and definitions

- System model:
  Ultimately a hierarchical object and interaction representation that's better than the graphical model, but that can be (continously, dynamically) simplified and used at an appropriate level, both for physics and graphics.

- Granularity of the system:
  Determins complexity. A car at a distance is a box with four wheels. A full fledged car simulator requires many more interacting objects and...

- Granularity of the interaction:
  Simple interaction may be used in some cases, and sometimes one switches to a more detailed model.

Classification and definitions

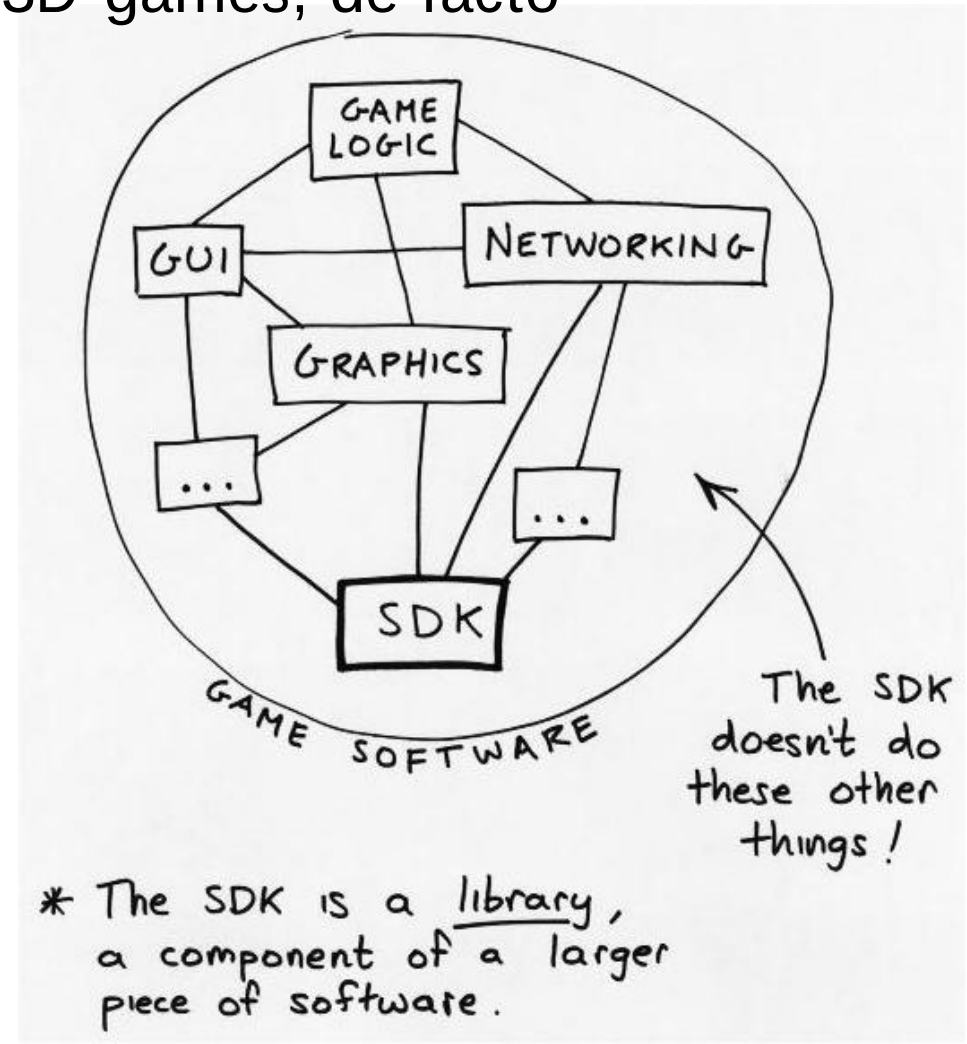
More about granularity of interaction and objects

In VR and for games physics is mainly used for simulating things that are visual. In addition the simulation must be on-line, interactive or even real-time. Granularity should be treated so that these constraints are obeyed.

In Science and Engineering, precision and accuracy are usually important. Off-line simulations are most common, e.g. FEM-lab in Matlab or plug-ins for CAD programs.

Architecture

Physics in 3D-games, de facto



Newton's laws

1. In an inertial frame, an object that is free of interaction has constant momentum.

2. The second law states the relationship between an acting force, inertia of a body, and its acceleration:

   $d\mathbf{p}/dt = \mathbf{F}$

   For point masses with constant mass ($\mathbf{p}=m\mathbf{v}$):

   $m\, d\mathbf{v}/dt = \mathbf{F}$ (or $\mathbf{F}=m\mathbf{a}$)

3. Each interaction force exerted by object A on object B has a corresponding reaction force of object B on object A. The reaction force is equal in magnitude but points in the opposite direction.

Newton's laws

Special case if the mass, m(t), is time dependent (raindrop, rocket, conveyor belt):

$$d\mathbf{p}(t)/dt = \mathbf{F}$$

$$d(m(t)\mathbf{v}(t))/dt = \mathbf{F}$$

Rocket Equation:

$$m\, d\mathbf{v}/dt - \mathbf{v}_e\, dm/dt = \mathbf{F}$$

where

$\mathbf{v}_e$ = exhaust velocity, $\mathbf{F}=m\mathbf{g}$ (gravitation g = 9.81m/s^2)

Dynamics of a particle

N2 for a particle

$\mathbf{F} = m\,\mathbf{a}$ or $\mathbf{F} = m\,d\mathbf{v}/dt$ or $F = m\,d^2\mathbf{x}/dt^2$

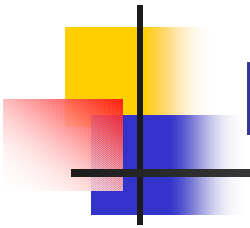
By integration

$\mathbf{v}(t+dt) = \mathbf{v}(t) + \mathbf{F}/m\ dt$

$\mathbf{x}(t+dt) = \mathbf{x}(t) + \mathbf{v}(t) + \frac{1}{2}\,\mathbf{F}/m\ dt^2$

or

$\mathbf{x}(t+dt) = \mathbf{x}(t) + \frac{1}{2}\,(\mathbf{v}(t) + \mathbf{v}(t+dt))\ dt$ (Euler's method $dt \rightarrow \Delta t$)



Numerical integration

Euler's method is not very accurate - often useless.

Alternatives: 2nd order Euler, Predictor-Corrector, Leap frog, Adams-Bashforth, Runge-Kutta, etc..

Large $\Delta t \Rightarrow$ fast time evolution but bad integration.

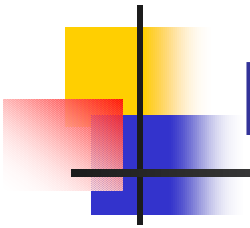
Small $\Delta t \Rightarrow$ slow time evolution and better integration.

If Δt too small $\Rightarrow$ numerical imprecision.

Choose Δt from experience, trial and error and "halfing" (start with a large timestep and half it until you are satisfied).

The system often gains energy and may eventually "explode" when using *explicit integration* (alt. *implicit integration*).

Not a great problem for interacting many body systems with stochastic force components (temperature, collisions, etc.).



Rigid Bodies

Can be seen as a collection of constrained particles (”rigid”).

”Integrate Newton’s laws for all these particles and their constraints”.

A. Sum up all linear forces and let them act on the centre of mass (CoM). Treat the center of mass as a point particle.

B. Sum up all angular forces to a net torque ($\mathbf{t} = \mathbf{r} \times \mathbf{F}$)

Both angular and linear momentum is conserved, unless a forces is exerted.

Rigid Bodies

For the CoM and linear momentum:

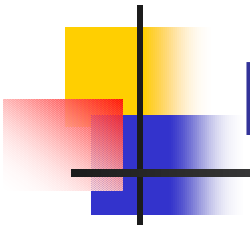
$$M\mathbf{V}(t) = \mathbf{F}$$

Define the angular momentum, $\mathbf{L}(t)$:

$\mathbf{L}(t) = \mathbf{I}(t)\boldsymbol{\omega}(t)$ ($\mathbf{I}$ = Inertia tensor, $\boldsymbol{\omega}$ = angular velocity)

N2 for rotation:

$d\mathbf{L}(t)/dt = \mathbf{I}\, d\boldsymbol{\omega}(t)/dt = \mathbf{I}\,\boldsymbol{\alpha}(t) = \boldsymbol{\tau}$ (torque) (if $\mathbf{I}$ constant)



Rigid Bodies

The Inertia tensor

3x3 matrix

The ”resistance” to angularly accelerating a body depends on its mass distribution around the axis of rotation. This is described by $\mathbf{I}$.

The mass distribution is described by the density, $\rho(\mathbf{x})$, of a body.

If the body is homogeneous (ρ constant), the Inertia tensor depends only on the geometry of the body.

The components of $\mathbf{I}$ are then obtained by simple volume integration.

Nature of forces and torques

- Acceleration due to gravity, $\mathbf{F} = m\,\mathbf{g}$
  where $\mathbf{g} = -g$ and $g = 9.81 m/s^2$.
- Other fields, e.g. an electron in a potential field. (All in all, there a four possible fields...)
- Engine, interaction,
- Friction, viscosity, air resistance, ...
- Elastic springs, Hooke's law, $\mathbf{F} = -k\,\mathbf{x}$
- Collisions (transfer of impulse – treated in a different way)
- Constraints

Damping forces

- Friction has electromagnetic origin from microscopic molecular and and interactions. Friction in mechanics is the resulting statistical effect.

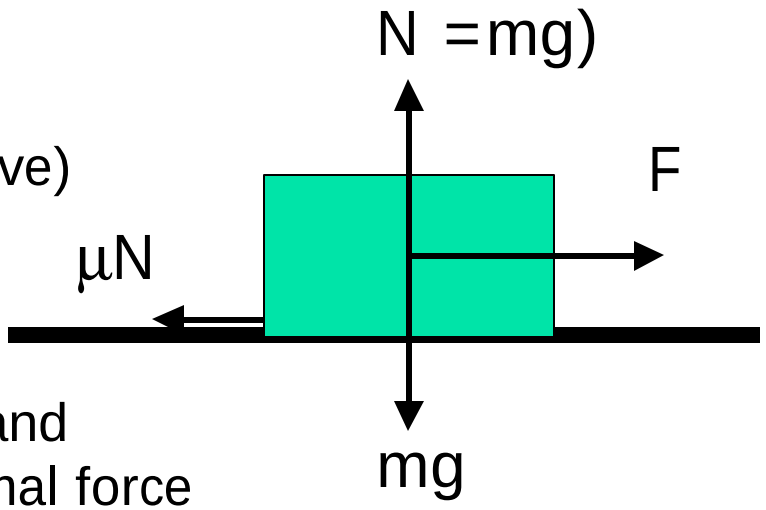
- Static friction (always reactive, ”sticking”)

  $F_f = -F$ if $F < F_{max}$
  $F_f = 0$ if $F > F_{max}$

- Dynamic friction (always reactive)

  $F_f = -\mu N$

  where μ is a friction constant and
  N is the magnitude of the normal force



Damping forces

- Viscosity (in fluids, water)

  $F_f = - K \eta v$ where η is a viscosity constant, and K a geometric constant ($K=6\pi R$ for a sphere, Stoke's law)

  Approximation for small velocities. At higher velocities the body disturbs the fluid in a non-linear way and more sofisticated models are needed.

- Air resistance

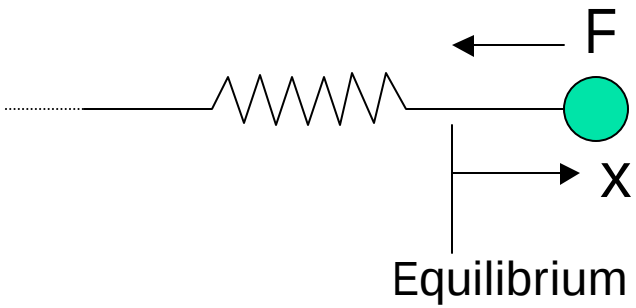
  $F_f = - c_v v^2$

  Also a linear approximation.
  N.B. the magnitude is proportional to v^2 but the direction points in the direct opposite direction of velocity, thus $\mathbf{F} = - c_v v^2 \mathbf{e}_v$

Spring models

Hooke's law:

$$F = - k x$$



k = spring constant

x = distance from equilibrium

Assume F = G(x)

Taylor expansion about x = 0:

$$G(x) = G(0) + G'(0)x + \tfrac{1}{2}\ G''(0)x^2 + O(x^3)$$

Spring models

At equilibrium $G(0) = 0$. For small x we can neglect $O(x^2)$

$G(x) \approx G'(0)\, x + O(x^2)$

If the force is restoring we can set $G'(0) = -k$ and thus we get Hooke's law:

$G(x) \approx -k\, x + O(x^2)$

Thus a spring model can represent MANY types of interaction close to equilibrium and is extremely versatile.

Spring models

N2 for the spring model:

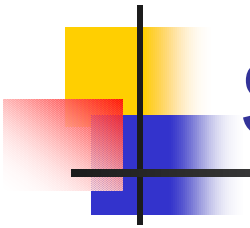
$m\, d^2x/dt^2 = - k\, x$

Can be solved analytically and gives

$x(t) = A \cos(\omega_0 t + \alpha)$ where $\omega_0^2 = k/m$

A and α are determined from initial conditions.

Harmonic oscillatory motion.



Spring models

Variants of the spring model (linear elastic model) :

- Soft tissue, rubber-type materials, bulk properties
- Bending rod, elastic rope
- Surfaces, water, cloth, fabric, skin

Collisions

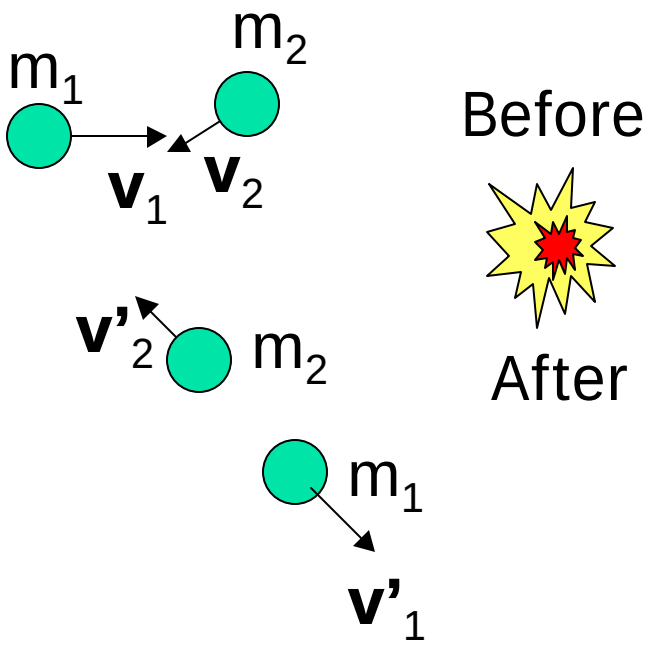
Two colliding particles

Energy conservation

$$E_{k1} + E_{k2} + E_{p1} + E_{p2} + E_i = E'_{k1} + E'_{k2} + E'_{p1} + E'_{p2} + E'_i$$

Momentum conservation

$$\mathbf{p}_1 + \mathbf{p}_2 = \mathbf{p}'_1 + \mathbf{p}'_2$$



Collisions

$E_k = mv^2$, $E_p = 0$ (for simplicity), E_i internal energy, $\mathbf{p}=m\mathbf{v}$

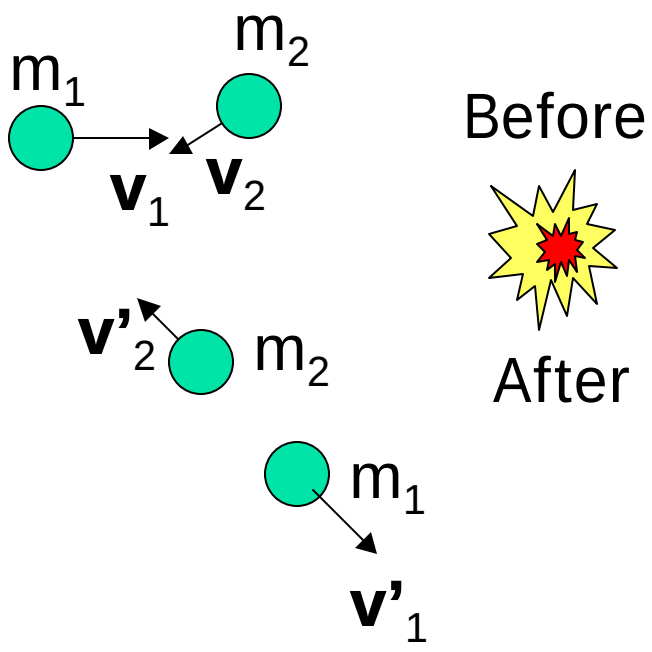
$Q = E'_i - E_i$ ($Q=0$ elastic, $Q<0$ endoergic, $Q>$ exoergic)

Energy conservation

$$m_1v_1^2 + m_2v_2^2 = m_1v'^2_1 + m_2v'^2_2 + Q$$

Momentum conservation

$$m_1\mathbf{v}_1 + m_2\mathbf{v}_2 = m_1\mathbf{v}'_1 + m_2\mathbf{v}'_2$$



An alternative to using Q, is to say that the initial kinetic energy is changed by a constant factor, the *restitution coefficient*.

The N-particle problem

Most mechanical problems for N=2 bodies can be solved analytically. For N>2 interacting bodies it can be proven (Poincaré, late 1800's) that it isn't possible to write down a closed general expression that describes the N-body system.

For N interacting bodies the problem grows as ½ N(N-1). With approximation methods and various types of speed-up schemes we can obtain ~ N logN.

Avogadro's number: ~10^{24} particles/Mole, thus ~10^{48} interactions per time step in a simulation...

Contacts and Collisions

Nature of a collision – extremely complicated – molecular electromagnetic forces...

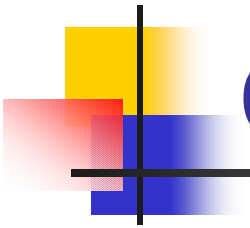
During a short time period, Δt, a force F(t) is acting.

This force changes the momentum, by transferring an *impulse*, $\Delta \mathbf{P}$,

$$\Delta \mathbf{P} = \int_{\Delta t} \mathbf{F}(t)dt$$ (details about **F**(t) are neglected)

P of the systems as a whole is still conserved.

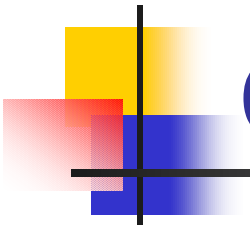
In simulations collisions are often treated by transferring impulse like this, *impulse trains*.



Contacts and Collisions

Handling a collision in a simulation

1. Objects are active, or deactivated - algorithms...
2. Collision engine detects a collision (and activates the objects) - algorithms...
3. Collision handling callback models the collision
   - Determine the exact collision time - algorithms...
   - Determine the volume / area of collision (vertices) – algorithms....
   - Model the collision, add impulse to the objects - algorithms...
4. Update / integrate the system - algorithms...
5. Goto 1.



Constraints

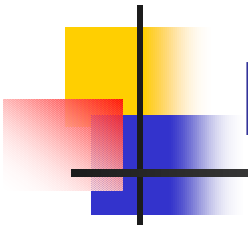
- Joints
- Hinges
- Interaction
- ”Rules”

- Handled using implicit or semi-implicit integration methods. See e.g. Baraff et.al.

Trouble

- Numerical instability, imprecision.
- Can’t find a solution to the Matrix equation and the Jacobian constraint problem, e.g. when there are many similar or redundant/conflicting constraints.

- Micro-macro problems.
  - A 10000kg box on top of a 0.1kg box, that stands on a solid surface.
  - A rotating rod that is 0.1m times 1000m.
  - Some objects move at 0.1m/s and some at 1000m/s (hard to detect collisions at all scales!).
- Object representation when objects crack or merge.



Fixes

- Adaptive time step
  - Δt depends on t
  - Δt depends on spatial region, objects and their states, and interaction.
- Avoid micro-macro problems...
- Cheat when solving the matrix equation – approximate eigenvalues better than none at all.
- Use cleaver ways of dynamically allocating objects – flat object hierarchy and dynamic trees.
- Alternatives to polygon mesh representations, e.g. implicit surfaces, primitives, fractals (?), ...

Some research trends

- ”Special FX”, fluids, cloth, clouds, ...
- Soft deformable matter
- Evolutionary dynamic design
  Genetic algorithms are used to design a system of interacting objects
- Space-time constraints
  How to teach an avatar to throw a basket ball that bounces seven times before giving three points?!
- Networked dynamics
  Dead-reckoning algorithms, extrapolation, smoothing, predictions
- Performance, parallelization, real-time aspects
- Architecture design: graphics – physics – rules – interaction – networking. Modularity and scalability.