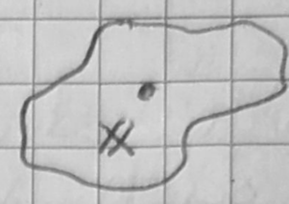
25 - Simulation in graphics

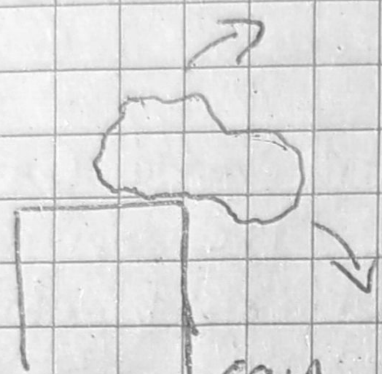
Particle vs rigid body



only has position, so 3 DOF

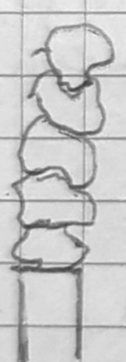
Has position and rotation, so 6 DOF

Collisions:



on collision it ~~will~~ can rotate, calculated much like last lecture, called impulse collision (a kind of).

It will vibrate if there is a small time difference on impact, need to set a "rest in contact" condition.



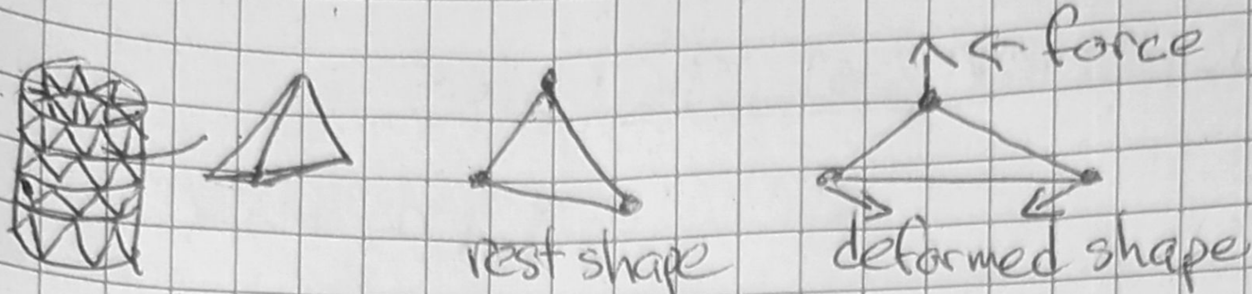
If we have several bodies with a "rest in contact" condition, we need to keep track of the stack this creates. This concept is called "stacking".

Articulated rigid body simulation:

If we look at the whole body, we have 6 DOF, but because it can deform at the joint, we have an additional 1 DOF. Used for ragdolls, vehicles etc.

Deformable object simulation:

A method is finite element method (FEM) where we use tetrahedralization, i.e. an object is built using tetrahedrons →

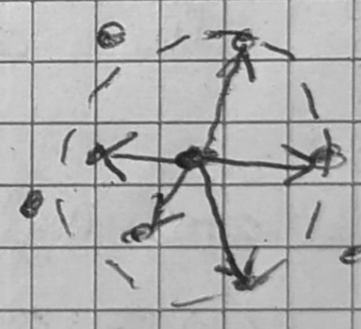


Each tetrahedron has a rest shape. When deformed each vertex has a force trying to get back to the rest shape.
This method is more favorable in some usecases today than mass-spring systems, because of higher fidelity.
like deformable objects,

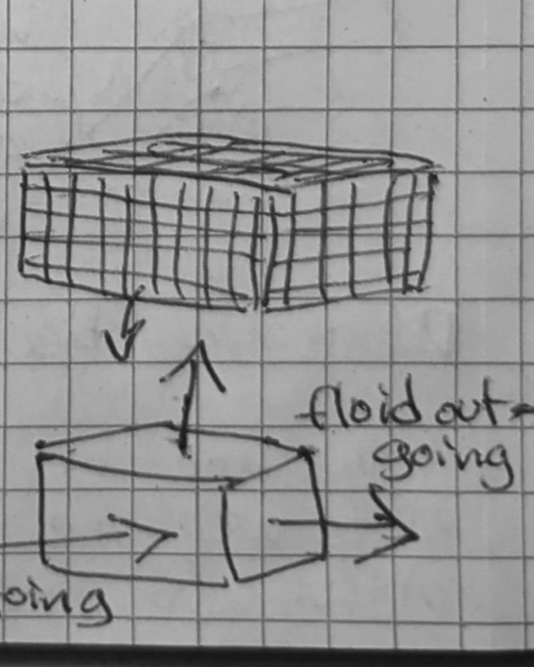
Mass-spring systems is still good for, for example, fracture simulation, where the spring are really stiff and breaks on impact, and cloth simulation. In cloth simulation we typically also have other kinds of spring forces between the verticies.

Fluid simulation

Particle-based/Lagrangian fluid simulation:
Particles interact with each other without springs, like fluid. We can wrap the system in a mesh to make it look like fluid.



One method for this is smoothed particle hydrodynamics (SPH), where we we check an area around each particle, and apply forces to the neighbouring particles using Navier-stokes equations.



An alternative is to use grid-based/Eulerian fuid simulation, where we split the volume into a grid, and for each cell we solve the Navier-stokes equations. The fluid coming into the cell faces should equal the fluid going out.

Hybrid fluid simulation:

Grid-based fluid simulation has the drawback that it can loose mass and volume, unlike particle based, while it have an easier time creating a divergent-free system (i.e. what comes into a cell must come out). We can use a hybrid system for this, where particle based calculates velocity for each particle and projects that to the grid system. The grid system then calculates the velocity field and sends that back.

Position-based dynamics (PBD):

distance constraint

Instead of internal forces
$f \rightarrow a \rightarrow \Delta v \rightarrow \Delta x$

Using position constraints
$C \longrightarrow \Delta x$

Newtonians dynamics using $F = ma$ can be unstable because of the time step being used. We can instead use position-based dynamics which directly manipulates positions ~~to~~ satisfy constraint

These are much more stable, and are becoming more popular.