Game Physics

Game and Media Technology
Master Program - Utrecht University

Dr. Nicolas Pronost
Soft body physics
Soft bodies

• In reality, objects are not purely rigid
  – for some it is a good approximation
  – but if you hit them with enough force, they will deform or break down

• In a game, you often want to see soft bodies (*i.e.* deformable objects)
  – car body, anything you punch or shoot at, *etc.*
  – piece of cloth, flag, paper sheet, *etc.*
  – snow, mud, lava, liquid, *etc.*
Elasticity

- **Elasticity** is the primary concept in soft body physics
- Property by which the body *returns to its original shape* after the forces causing the deformation are removed
  - A plastic rod can easily be bended, and returned to its original form
  - A steel rod is difficult to bend, but can also return to its original form
Stress

• The **stress** within an object is the magnitude of an applied force divided by the area of its application
  – large value when the force is large or when the surface is small

• It is a pressure measure $\sigma$ and has the unit Pascal $Pa = N/m^2$

• Example
  – the stress on the plane is $\sigma = mg/\pi r^2$
Strain

• The **strain** on an object $\varepsilon$ is the fractional deformation caused by a stress
  – dimensionless (change in dimension relative to original dimension)
  – measures how much a deformation differs from a rigid body transformation
    • negative if compression, zero if rigid body transformation, positive if stretch

• Example
  – the strain on the rod is $\varepsilon = \Delta L / L$
Body material

- Stress and strain do not contain information about the specific material (i.e. deformation behavior) to which a force is applied.
- The amount of stress to produce a strain does.
- Therefore, we can model it by the ratio of stress to strain:
  - usually in a linear direction, along a planar region or throughout a volume region
    - Young’s modulus, Shear modulus, Bulk modulus
  - they describe the different ways the material changes shape due to stress.
Young’s modulus

- The **Young’s modulus** is defined as the ratio of linear stress to linear strain

\[ Y = \frac{\text{linear stress}}{\text{linear strain}} = \frac{F/A}{\Delta L/L} \]

- Example
Shear modulus

- The **Shear modulus** is defined as the ratio of planar stress to planar strain

\[ S = \frac{\text{planar stress}}{\text{planar strain}} = \frac{F/A}{\Delta L/L} \]

- Example
Bulk modulus

- The **Bulk modulus** is defined as the ratio of volume stress to volume strain (inverse of compressibility)

\[
B = \frac{\text{volume stress}}{\text{volume strain}} = \frac{\Delta P}{\Delta V/V}
\]

- Example
Poisson’s ratio

• The Poisson’s ratio is the ratio of transverse to axial strain

\[ \nu = - \frac{d \text{ transverse strain}}{d \text{ axial strain}} \]

– negative transverse strain in axial tension, positive in axial compression
– negative axial strain in compression, positive in tension
– equals 0.5 in perfectly incompressible material

• If the force is applied along \( x \) then we have

\[ \nu = - \frac{d \varepsilon_y}{d \varepsilon_x} = - \frac{d \varepsilon_z}{d \varepsilon_x} \]
Poisson’s ratio

• Example of a cube of size $L$

![Diagram showing a cube deformed under force $F$]

\[
\begin{align*}
\Delta L' &= \Delta L + \nu \Delta L \\
1 - \nu &= 1 + \frac{\Delta L'}{L} \\
\nu &\approx \frac{\Delta L'}{\Delta L}
\end{align*}
\]
Continuum mechanics

• A deformable object is defined by its rest shape and the material parameters
• In the discrete case, the object $M$ is a discrete set of points with material coordinates $m \in M$ that samples the rest shape of the object
• When forces are applied, the object deforms
  – each $m$ moves to a new location $x(m)$
  – $u(m) = x(m) - m$ can be seen as the displacement vector field
  – e.g. a constant displacement field is a translation of the object
Continuum mechanics

- Material coordinate $P$ with position $X$ is deformed to $p$ with position $x$
- Material coordinate $Q$ with position $X + dX$ is deformed to $q$ with position $x + dx$
- If the deformation is very small (i.e. linear deformation in interval $\Delta t$), the displacements of the material coordinates can be described by

$$\begin{align*}
x + dx &= X + dX + u(X + dX) \\
dx &= X - x + dX + u(X + dX) \\
dx &= dX + u(X + dX) - u(X) \\
dx &= dX + du
\end{align*}$$
Continuum mechanics

\[ u(X + dX) = u(X) + du \]

\[ dx = dX + du \]
• $du$ is the **relative displacement vector**

• It represents the relative displacement of $Q$ with respect to $P$ in the deformed configuration

• Now if we assume that $Q$ is very close to $P$ and that the displacement field is continuous, we have

$$u(X + dX) = u(X) + du \approx u(X) + \nabla u \ast dX$$

where the gradient of the displacement field is (in 3D) the $3 \times 3$ matrix of the partial derivatives of $u$

$$\nabla u = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}$$

where $u = (u, v, w)^T$
Continuum mechanics

• With that definition of the relative displacement vector, we can calculate the relative position of $q$

$$dx = dX + du = dX + \nabla u \ast dX$$

$$dx = (I + \nabla u)dX = F \ast dX$$

• We call $F$ the material deformation gradient tensor

• It characterizes the local deformation at a material coordinate, i.e. provides a mapping between the relative position at rest and the relative position after deformation
Strain and stress

- The strain and stress are related to the material deformation gradient tensor $F$, and so to the displacement field $u$

- In interactive applications, we usually use the Green-Cauchy strain tensors

$$\epsilon_G = \frac{1}{2} (\nabla u + (\nabla u)^T + (\nabla u)^T \nabla u)$$

$$\epsilon_C = \frac{1}{2} (\nabla u + (\nabla u)^T)$$

- And stress tensor from Hooke’s linear material law

$$\sigma = E \ast \epsilon$$

where $E$ is the elasticity tensor and depends on the Young’s modulus and Poisson’s ratio (and more)
Modeling soft bodies

- Two types of approaches are possible to simulate deformable models
  - Lagrangian methods (particle-based)
    - a model consists of a set of moving points carrying material properties
    - convenient to define an object as a connected mesh of points or a cloud of points, suitable for deformable soft bodies
    - examples: Finite Element/Difference/Volume methods, Mass-spring system, Coupled particle system, Smoothed particle hydrodynamics
  - Eulerian methods (grid-based)
    - scene is a stationary set of points where the material properties change over time
    - boundary of object not explicitly defined, suitable for fluids
Finite Element Method

• FEM is used to numerically solve partial differential equations (PDEs) by discretization of the volume into a large finite number of disjoint elements (3D volumetric mesh)

• The PDE of the equation of motion governing dynamic elastic materials is given by

\[ \rho \ast \alpha = \nabla \cdot \sigma + F \]

where \( \rho \) is the density of the material, \( \alpha \) is the acceleration of the element, \( \nabla \cdot \sigma \) is the divergence of stress (internal forces) and \( F \) the external forces.
Finite Element Method

- First the deformation field $u$ is estimated from the positions of the elements within the object
- Given the current local strain, the local stress is calculated
- The equation of motion of the element nodes is obtained by integrating the stress field over each element and relating this to the node accelerations through the deformation energy

$$E = \int_V \epsilon(m) \ast \sigma(m) \, dm$$
Finite Differences Method

• If the object $M$ is sampled using a regular spatial grid, the PDE can be discretized using finite differences (FD)
  – easier to implement than FEM
  – difficult to approximate complex boundaries

• Deformation energy comes from difference between metric tensors of the deformed and original shapes

• Derivative of this energy is discretized using FD

• Finally semi-implicit integration is used to move forward through time
Finite Volume Method

• In the Finite Volume method, the nodal forces are not calculated from the derivation of the deformation energy
• But first internal forces $f$ per unit area of a plane (of normal $n$) are calculated from the stress tensor
  \[ f = \sigma \cdot n \]
• The total force acting on a face $A$ of an element is
  \[ f_A = \int_A \sigma \, dA = A \cdot \sigma \cdot n \]
  for planar element faces (stress tensor constant within an element)
• By iterating on all faces of an element, we can then distribute (evenly) the force among adjacent nodes
Boundary Element Method

• The boundary element method simplifies the finite element method from a 3D volume problem to a 2D surface problem
  – PDE is given for boundary deformation
  – only works for homogenous material
  – topological changes more difficult to handle
Mass-Spring System

- An object consists of point masses connected by a network of massless springs.
- The state of the system is defined by the positions $x_i$ and velocities $v_i$ of the masses $i = 1 \ldots n$.
- The force $f_i$ on each mass is computed from the external forces (e.g. gravity, friction) and the spring connections with its neighbors.
- The motion of each mass point $f_i = m_i a_i$ is summed up for the entire system in
  $$M \ast a = f(x, v)$$
where $M$ is a $3n \times 3n$ diagonal matrix.
Mass-Spring System

- The mass points are usually regularly spaced in a 3D lattice
- The 12 edges are connected by structural springs
  - resist longitudinal deformations
- Opposite corner mass points are connected by shear springs
  - resist shear deformations
- The rest lengths define the rest shape of the object
Mass-Spring System

• The force acting on mass point $i$ generated by the spring connecting $i$ and $j$ is

$$f_i = Ks_i(|x_{ij}| - l_{ij}) \frac{x_{ij}}{|x_{ij}|}$$

where $x_{ij}$ is the vector from positions $i$ to $j$, $K_i$ is the stiffness of the spring and $l_{ij}$ is the rest length.

• To simulate dissipation of energy along the distance vector, a damping force is added

$$f_i = Kd_i \left( \frac{(v_j - v_i)^T x_{ij}}{x_{ij}^T x_{ij}} \right) x_{ij}$$
Mass-Spring System

• Intuitive system and simple to implement
• Not accurate as does not necessarily converge to correct solution
  – depends on the mesh resolution and topology
  – spring constants chosen arbitrarily
• Can be good enough for games, especially cloth animation
  – as can have strong stretching resistance and weak bending resistance
Coupled Particle System

- Particles interact with each other depending on their spatial relationship
- Referred to as spatially coupled particle system
  - these relationships are dynamic, so geometric and topological changes can take place
- Each particle $p_i$ has a potential energy $E_{Pi}$ which is the sum of the pairwise potential energies between the particle $p_i$ and the other particles

$$E_{Pi} = \sum_{j \neq i} E_{Pij}$$
Coupled Particle System

• The force $f_i$ applied on the particle at position $p_i$ is

$$f_i = -\nabla p_i E_{Pi} = - \sum_{j \neq i} \nabla p_i E_{Pij}$$

where $\nabla p_i E_{Pi} = \left( \frac{dE_{Pi}}{dx_i}, \frac{dE_{Pi}}{dy_i}, \frac{dE_{Pi}}{dz_i} \right)$

• To reduce computational costs, interactions to a neighborhood is used
  – potential energies weighted according to distance to particle
Smoothed Particle Hydrodynamics

• SPH uses discrete particles to compute approximate values of needed physical quantities and their spatial derivatives
  – obtained by a distance-weight sum of the relevant properties of all the particles which lie within the range of a smoothing kernel

• Reduces the programming and computational complexity
  – suitable for gaming applications
Smoothed Particle Hydrodynamics

• The equation for any quantity \( A \) at any point \( r \) is given by

\[
A(r) = \sum_j m_j \frac{A_j}{\rho_j} W(|r - r_j|, h)
\]

– where \( W \) is the smoothing kernel (usually Gaussian function or cubic spline) and \( h \) the smoothing length (max influence distance)
– for example the density can be calculated as

\[
\rho(r) = \sum_j m_j W(|r - r_j|, h)
\]

• It is applied to pressure and viscosity forces, while external forces are applied directly to the particles
Smoothed Particle Hydrodynamics

• The spatial derivative of a quantity can be calculated from the gradient of the kernel
  – the equations of motion are solved by deriving forces

• By varying automatically the smoothing length of individual particles you can tune the resolution of a simulation depending on local conditions
  – typically use a large length in low particle density regions and a smaller length in high density regions

• Easy to conserve mass (constant number of particles) but difficult to maintain incompressibility of the material
Eulerian Methods

• Eulerian methods are typically used to simulate fluids (liquids, smoke, lava, cloud, etc.)
• The scene is represented as a regular voxel grid, and fluid dynamics describes the displacements
  – we apply finite difference formulation on the voxel grid
  – the velocity is stored on the cell faces and the pressure is stored at the center of the cells
• Heavily rely on the Navier-Stokes equations of motion for a fluid
Navier-Stokes equations

They represent the conservation of mass and momentum for an incompressible fluid

\[ \nabla \cdot \mathbf{u} = 0 \]

\[ \rho ( \mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} ) = \nabla \cdot ( \nu \nabla \mathbf{u} ) - \nabla p + f \]

- \( \mathbf{u}_t \) is the time derivative of the fluid velocity (the unknown), \( p \) is the pressure field, \( \nu \) is the kinematic viscosity, \( f \) is the body force per unit mass (usually just gravity \( \rho g \))
Navier-Stokes equations

• First $f$ is scaled by the time step and added to the current velocity
• Then the advection term $u \cdot \nabla u$ is solved
  – it governs how a quantity moves with the underlying velocity field (time independent, only spatial effect)
  – it ensures the conservation of momentum
  – sometimes called convection or transport
  – solved using a semi-Lagrangian technique
Navier-Stokes equations

- Then the viscosity term $\nabla \cdot (\nu \nabla u) = \nu \nabla^2 u$ is solved
  - it defines how a cell interchanges with its neighbors
  - also referred to as diffusion
  - viscous fluids can be achieved by applying diffusion to the velocity field
  - it can be solved for example by finite difference and an explicit formulation

  - 2-neighbor 1D:
    \[ u_i(t) = \nu \times \Delta t \times (u_{i+1} + u_{i-1} - 2u_i) \]
  - 4-neighbor 2D:
    \[ u_{i,j}(t) = \nu \times \Delta t \times (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) \]

- Taking the limit gives indeed $\nu \nabla^2 u$
Navier-Stokes equations

• Finally, the pressure gradient is found so that the final velocity will conserve the volume (i.e. mass for incompressible fluid)
  – sometimes called pressure projection
  – it represents the resistance to compression $\nabla p$
Navier-Stokes equations

- We make sure the velocity field stays divergence-free with the second equation $\nabla \cdot u = 0$, i.e. the velocity flux of all faces at each fluid cell is zero (everything that comes in, goes out)

- The equation $u(t + \Delta t) = u(t) - \Delta t \nabla p$ is solved from its combination with $\nabla \cdot u = 0$, giving
  \[
  \nabla \cdot u(t + \Delta t) = \nabla \cdot u(t) - \Delta t \nabla \cdot (\nabla p) = 0
  \]
  \[
  \iff \Delta t \nabla^2 p = \nabla \cdot u(t)
  \]
with which we solve for $p$, then plug back in the $u(t + \Delta t)$ equation to calculate the final velocity
Navier-Stokes equations

- Compressible fluids can also conserve mass, but their density must change to do so
- Pressure on boundary nodes
  - In free surface cells, the fluid can evolve freely \((p = 0)\)
    - so that for example a fluid can splash into the air
  - Otherwise (e.g. in contact with a rigid body), the fluid cannot penetrate the body but can flow freely in tangential directions \(u_{\text{boundary}} \cdot n = u_{\text{body}} \cdot n\)
End of
Soft body physics

Next
Physics engine design and implementation