Computer Animation & Physical Simulation

Lecture 7: Rigid-Body Simulation II

XIAOPEI LIU
School of Information Science and Technology
ShanghaiTech University

I. Review of Basic Rigid Body Simulation

Non-constrained Rigid Body Simulation

- **Time rate change of the state variable**

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t)^{*}R(t) \\ F(t) \\ \tau(t) \end{pmatrix}$$

- **Computing order**

$$F(t) \longrightarrow P(t) \qquad \tau(t) \longrightarrow L(t)$$

$$P(t) \xrightarrow{P(t) = Mv(t)} v(t) \longrightarrow x(t) \qquad L(t) \xrightarrow{L(t) = I(t)\omega(t)} \omega(t) \longrightarrow R(t)$$

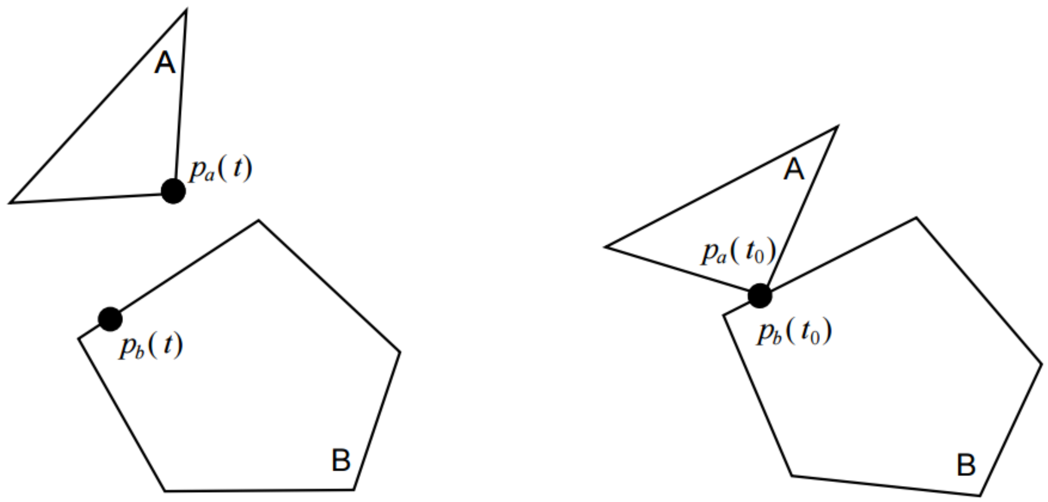
Non-constrained Rigid Body Simulation

- **Runge–Kutta methods** $y' = f(x, y)$
  - Achieve higher accuracy
  - Re-evaluate f(·, ·) at points intermediate between (x_n, $y(x_n)$) and (x_n+1, $y(x_n+1)$)

$$
\begin{aligned}
y_{n+1} &= y_n + h\Phi(x_n, y_n; h)\ , \\
\Phi(x, y; h) &= \sum_{r=1}^{R} c_r k_r\ , \\
k_1 &= f(x, y)\ , \\
k_r &= f\left(x + ha_r, y + h\sum_{s=1}^{r-1} b_{rs} k_s\right)\ , \quad r = 2, \ldots, R\ , \\
a_r &= \sum_{s=1}^{r-1} b_{rs}\ , \quad r = 2, \ldots, R\ .
\end{aligned}
$$

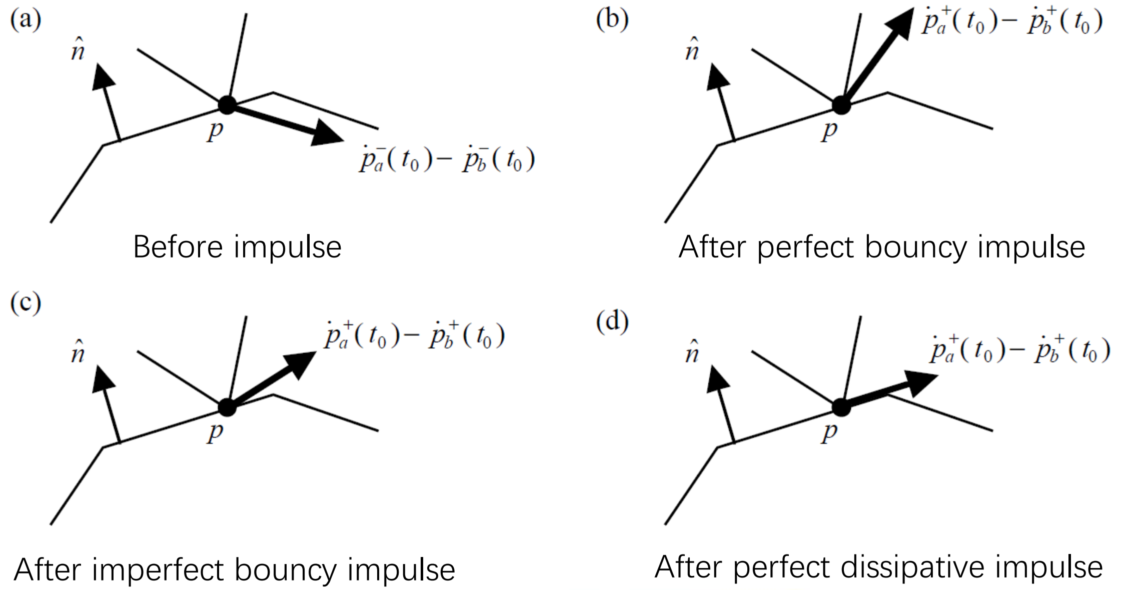
Constrained Rigid Body Simulation

- **Colliding contact**



Constrained Rigid Body Simulation

- **Physical meaning for coefficient of restitution**
  - Illustration



Constrained Rigid Body Simulation

- **Colliding contact**
  - How to compute the impulse?
  - For frictionless bodies, the direction of the impulse will be in the normal direction

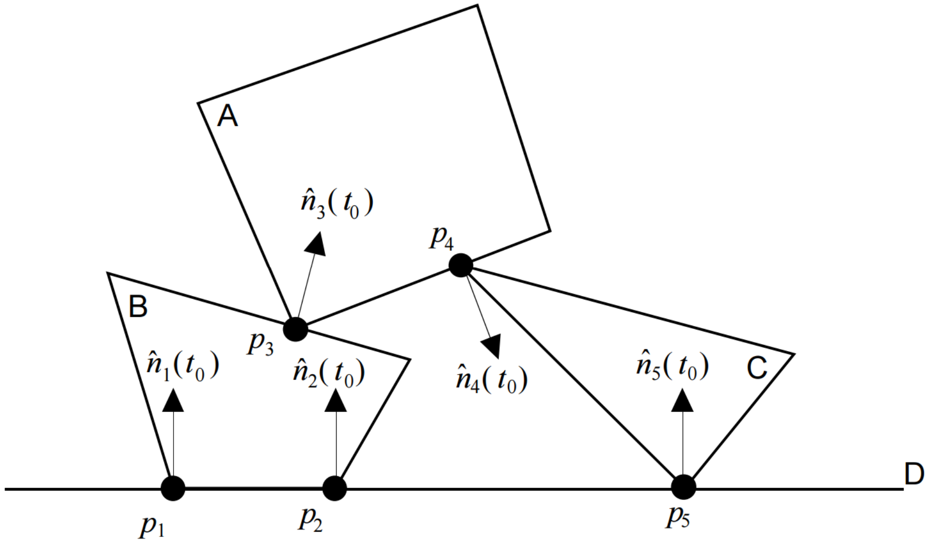
$$J = j\hat{n}(t_0)$$

$$j = \frac{-(1+\epsilon)v_{rel}^-}{\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0) \cdot \left(I_a^{-1}(t_0)\left(r_a \times \hat{n}(t_0)\right)\right) \times r_a + \hat{n}(t_0) \cdot \left(I_b^{-1}(t_0)\left(r_b \times \hat{n}(t_0)\right)\right) \times r_b}$$

$$v_{rel}^+ = -\epsilon v_{rel}^- \qquad 0 \leq \epsilon \leq 1$$

Constrained Rigid Body Simulation

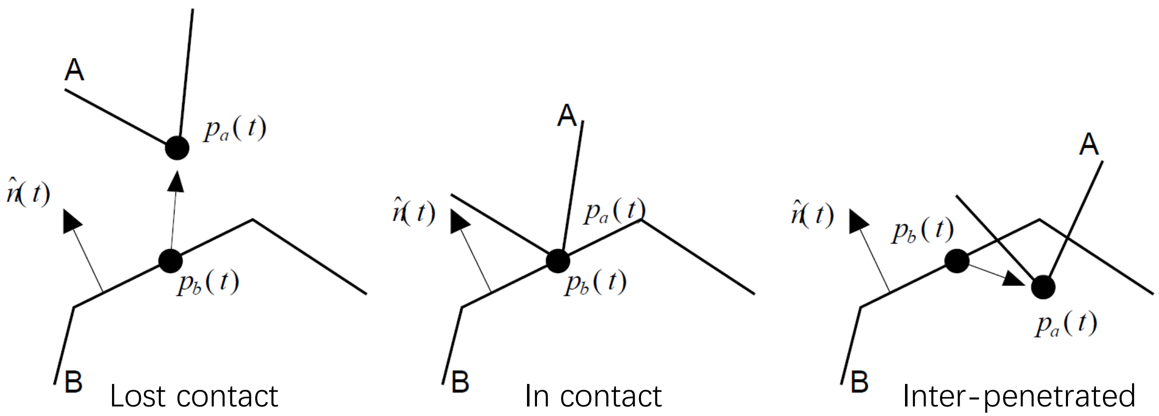
- **Resting contact**
  - Computing contact forces



Constrained Rigid Body Simulation

- **Resting contact**
  - Preventing inter-penetration
    - Construction of separation distance

$$d_i(t) = \hat{n}_i(t) \cdot (p_a(t) - p_b(t))$$



Constrained Rigid Body Simulation

- **Resting contact**
  - Preventing inter-penetration
    - Consider $d_i(t_0) = 0$
    - We have to keep the two bodies from accelerating towards each other
      - Taking the second derivative of $d_i(t_0)$

$$\ddot{d}(t_0) = \hat{n}_i(t_0) \cdot (\ddot{p}_a(t_0) - \ddot{p}_b(t_0)) + 2\dot{\hat{n}}_i(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

      - $\ddot{d}_i(t_0) > 0$: contact will break immediately
      - $\ddot{d}_i(t_0) = 0$: contact remains
      - $\ddot{d}_i(t_0) < 0$: must be avoided

II. Mathematical Formulation of Constraints

Resting Contact

- **Expression for three conditions**
  - Non-interpenetration

$$\ddot{d}_i(t_0) \geq 0$$

  - Repulsiveness

$$f_i \geq 0$$

  - Contact breaking

$$f_i \ddot{d}_i(t_0) = 0$$

Resting Contact

- **Computing contact force**
  - Force contribution
    - Consider the expression

$$\ddot{d}(t_0) = \hat{n}_i(t_0) \cdot (\ddot{p}_a(t_0) - \ddot{p}_b(t_0)) + 2\dot{\hat{n}}_i(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

$$\ddot{p}_a(t) = \dot{v}_a(t) + \dot{\omega}_a(t) \times r_a(t) + \omega_a(t) \times (\omega_a(t) \times r_a(t))$$

    - Forces contribute to
      - Linear acceleration $\dot{v}_a(t)$
      - Angular acceleration $\dot{\omega}_a(t) = I_a^{-1}(t)\tau_a(t) + I_a^{-1}(t)(L_a(t) \times \omega_a(t))$

Resting Contact

- **Computing contact force**
  - Express separating distance acceleration in terms of all associated forces

$$\ddot{d}_i(t_0) = a_{i1} f_1 + a_{i2} f_2 + \cdots + a_{in} f_n + b_i$$

  - Write for all contact points

$$\begin{pmatrix} \ddot{d}_1(t_0) \\ \vdots \\ \ddot{d}_n(t_0) \end{pmatrix} = \mathbf{A} \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} + \begin{pmatrix} b_i \\ \vdots \\ b_n \end{pmatrix}$$

Resting Contact

- **Computing contact force**
  - Contribution of a_{ij}
    - Consider linear and angular acceleration

$$\frac{\hat{n}_j(t_0)}{m_a} + \left(I_a^{-1}(t_0)\left((p_j - x_a(t_0)) \times \hat{n}_j(t_0)\right)\right) \times r_a$$

  - Contribution of b_i
    - Collect the force independent part

$$2\dot{\hat{n}}_i(t_0) \cdot (\dot{p}_a(t_0) - \dot{p}_b(t_0))$$

$$\frac{F_a(t_0)}{m_a} + \left(I_a^{-1}(t_0)\tau_a(t_0)\right) \times r_a + \omega_a(t_0) \times (\omega_a(t_0) \times r_a) + \left(I_a^{-1}(t_0)(L_a(t_0) \times \omega_a(t_0))\right) \times r_a$$

Contact Formulation

- **Mathematical Formulation**

$$\mathbf{a} = \mathbf{A}\boldsymbol{f} + \mathbf{b} \qquad a_i \geq 0, \;\; f_i \geq 0 \quad \text{and} \quad f_i a_i = 0$$

$$\sum_{i=1}^{n} f_i a_i = \boldsymbol{f}^T \mathbf{a} = 0$$

$$\mathbf{A}\boldsymbol{f} + \mathbf{b} \geq \mathbf{0}, \;\; \boldsymbol{f} \geq \mathbf{0} \quad \text{and} \quad \boldsymbol{f}^T(\mathbf{A}\boldsymbol{f} + \mathbf{b}) = 0.$$

Contact Formulation

- **Mathematical Formulation**
  - Linear complementarity problem

$$\mathbf{A}\boldsymbol{f} + \mathbf{b} \geq \mathbf{0}, \quad \boldsymbol{f} \geq \mathbf{0} \qquad \text{and} \qquad \boldsymbol{f}^T(\mathbf{A}\boldsymbol{f} + \mathbf{b}) = 0.$$

$$\min_{\boldsymbol{f}} \boldsymbol{f}^T(\mathbf{A}\boldsymbol{f} + \mathbf{b}) \qquad \text{subject to} \qquad \left\{ \begin{array}{c} \mathbf{A}\boldsymbol{f} + \mathbf{b} \geq \mathbf{0} \\ \boldsymbol{f} \geq \mathbf{0} \end{array} \right\}$$

Solution of Linear Complementarity Problem

- **Minimum Map Newton Method**

$$y = ax + b,$$
$$y \geq 0, x \geq 0, \text{ and } xy = 0$$

$$\Rightarrow \quad h(x, y) = \min(x, y) \qquad \min(x, y) = \begin{cases} x & \text{if } x < y \\ y & \text{otherwise} \end{cases}$$

$$h(x^*, y^*) = 0 \quad \text{iff} \quad 0 \leq y^* \quad \perp \quad x^* \geq 0$$

$$\mathbf{H}(\mathbf{x}) = \mathbf{H}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} h(\mathbf{x}_1, \mathbf{y}_1) \\ \dots \\ h(\mathbf{x}_n, \mathbf{y}_n) \end{bmatrix} = \mathbf{0}$$

Solution of Linear Complementarity Problem

- **Fischer–Newton Method**

$$y = ax + b,$$
$$y \geq 0, x \geq 0, \text{ and } xy = 0$$

$$\phi(x, y) \equiv \sqrt{x^2 + y^2} - x - y$$

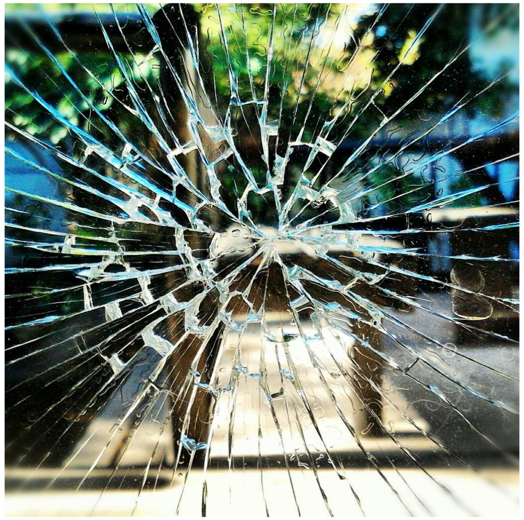
$$\phi(x^*, y^*) = 0 \quad \text{iff} \quad 0 \leq y^* \quad \perp \quad x^* \geq 0$$

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{y}_1) \\ \vdots \\ \phi(\mathbf{x}_n, \mathbf{y}_n) \end{bmatrix} = \mathbf{0}$$

IV. Rigid Body Fracture

What is a fracture?

- **A fracture**
  - The separation of an object or material into two or more pieces under the action of stress



How to model fracture?

- **Consider material deformations**
  - Even rigid body objects have small deformations
  - Deformation causes change of internal stress
  - Fracture arises when internal stress exceed the material toughness (strength)

- **Modeling**
  - Computation of internal stress distribution
  - Determine the fracture point and fracture geometry

Continuum Mechanics

- **A branch of mechanics**
  - Modeled as a continuous mass rather than as discrete particles
  - The matter in the body is continuously distributed
  - A continuum is a body that can be continually sub-divided into infinitesimal elements
    - Derivatives are available to compute
  - Deal with deformable bodies
    - As opposed to ideal rigid bodies
    - Analyzing internal force of rigid bodies should consider deformation (very small)

Material Coordinates

- **Eulerian specification of a field**
  - Represented as a function of position **x** and time *t*

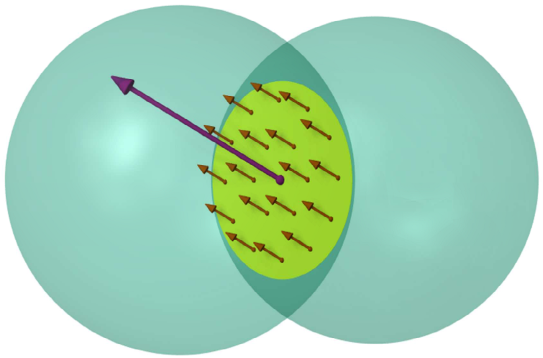
$$\mathbf{u}\left(\mathbf{x}, t\right)$$

- **Lagrangian (material) specification**
  - Particles are followed through time
  - Particles are labeled by some (time-independent) vector field $\mathbf{x_0}$ (material coordinates)

$$\mathbf{u}\left(\mathbf{X}(\mathbf{x}_0, t), t\right) = \frac{\partial \mathbf{X}}{\partial t}\left(\mathbf{x}_0, t\right)$$

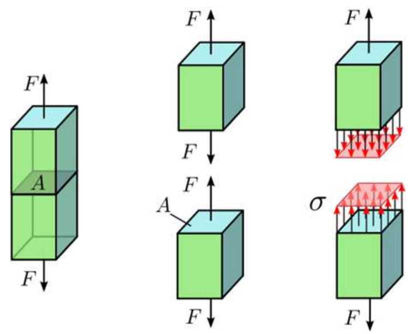
Stress of a Material

- **A physical quantity of material**
  - Internal forces that neighboring particles exert on each other
  - Defined as the force across a "small" boundary per unit area of that boundary

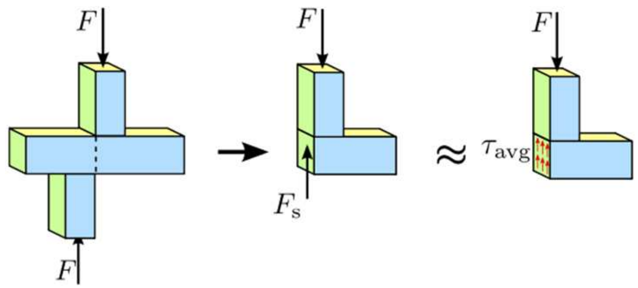


Stress of a Material

- **Stress may be regarded as the sum of two components**
  - Normal stress
    - The stress component perpendicular to the surface (compression or tension)
  - Shear stress
    - The stress component parallel to the surface



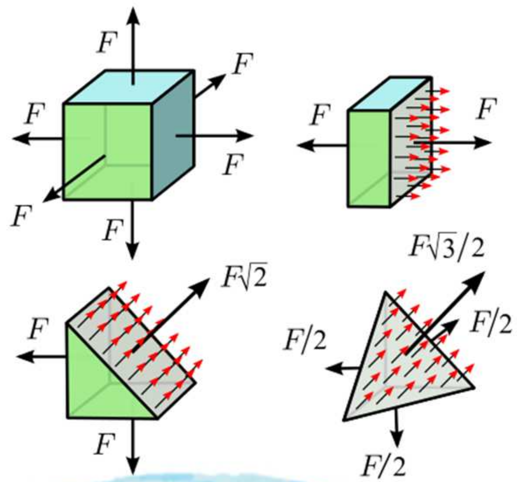
Normal stress



Shear stress

Stress of a Material

- **Isotropic stress**
  - A simple type of stress
  - Occur when the material body is under equal compression or tension in all directions
  - The material is homogeneous



Cauchy Stress Tensor

- **General stress**
  - Mechanical bodies experience more than one type of stresses at the same time (combined stress)
  - Combined stresses cannot be described by a single vector

- **Cauchy’s observation**
  - The stress vector across a surface will always be a linear function of the surface's normal

$$T = \boldsymbol{\sigma}(n) \qquad\qquad \boldsymbol{\sigma}(\alpha u + \beta v) = \alpha\boldsymbol{\sigma}(u) + \beta\boldsymbol{\sigma}(v)$$

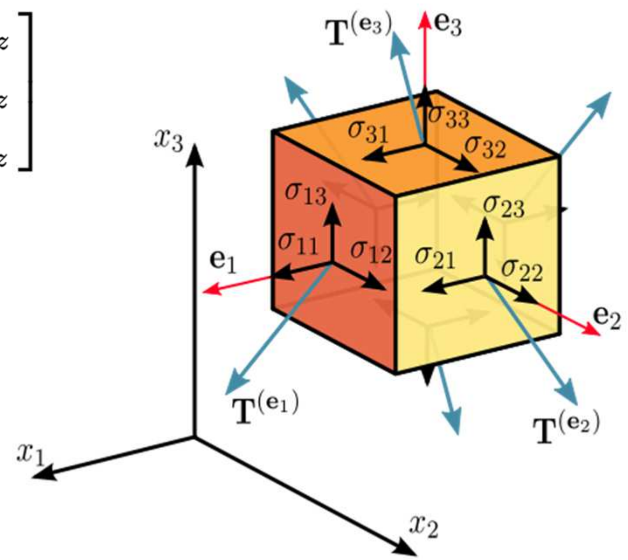
Cauchy Stress Tensor

- **Definition**
  - A 3x3 matrix

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

$$T = n \cdot \boldsymbol{\sigma}$$

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$



Cauchy Stress Tensor

- **Symmetric stress tensor**
  - Conservation of angular momentum implies that the stress tensor is symmetric

$$\begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix}$$

    - Normal stresses

$$\sigma_x, \sigma_y, \sigma_z$$

    - Shear stresses

$$\tau_{xy}, \tau_{xz}, \tau_{yz}$$

Deformation of Material

- **Physical view of deformation**
  - Transformation of a body from a reference configuration to the current configuration
  - A configuration is a set containing the positions of all particles of the body

- **Causes of deformation**
  - External loads (usually on the exterior surfaces)
  - Body forces (volumetric force within the whole body, e.g., gravity force)

Types of Deformation

- **Elastic deformations**
  - Deformations are recovered after the stress field has been removed

- **Irreversible deformation**
  - Deformations remain even after stresses have been removed
  - Plastic deformation
    - Occurs in material bodies after stresses have attained a certain threshold value (elastic limit or yield stress)

Strain of Material

- **What is a strain**
  - A description of deformation in terms of relative displacement of particles
  - The relation between stresses and induced strains is expressed by constitutive equations
    - E.g., Hooke's law for linear elastic materials
- **Formulation**
  - A general deformation of a body can be expressed in the form

$$\mathbf{x} = \boldsymbol{F}(\mathbf{X})$$

    - **X** is the reference position of material points in the body

Strain of Material

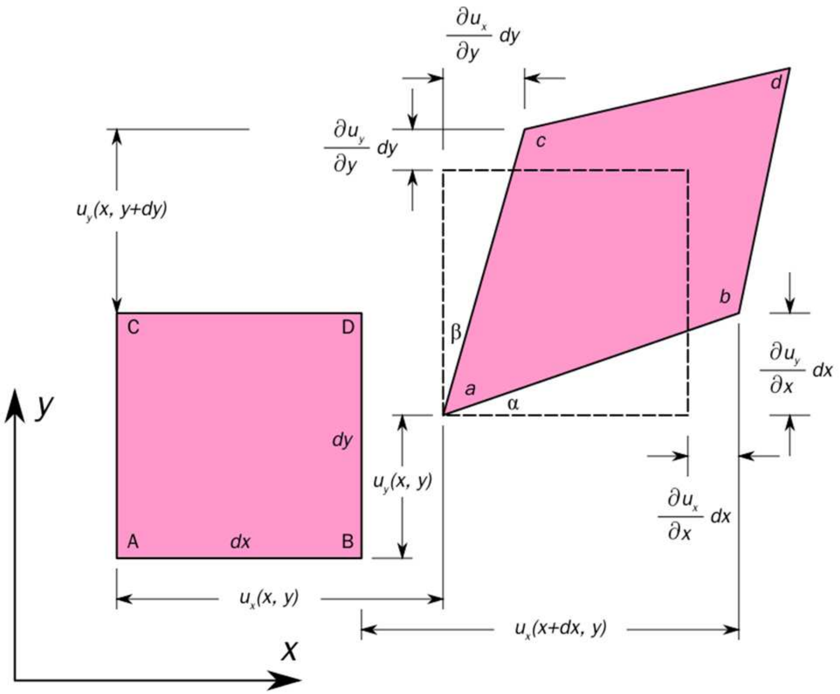
- **Formulation**
  - Such a measure does not distinguish between rigid body motions and changes in shape of the body
  - Mathematical definition

$$\boldsymbol{\varepsilon} \doteq \frac{\partial}{\partial \mathbf{X}} (\mathbf{x} - \mathbf{X}) = \boldsymbol{F}' - \boldsymbol{I}$$

    - Strains measure how much a given deformation differs locally from a rigid-body deformation

Strain of Material

- **Normal and shear strain**
  - A normal strain is perpendicular to the face of an element
  - A shear strain is parallel to it
  - Consistent with normal stress and shear stresses



Strain of Material

- **Strain tensor**

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \qquad \varepsilon_{yy} = \frac{\partial u_y}{\partial y} \qquad \varepsilon_{zz} = \frac{\partial u_z}{\partial z}$$

$$\gamma_{xy} = \gamma_{yx} = \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y}$$

$$\gamma_{yz} = \gamma_{zy} = \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}$$

$$\gamma_{zx} = \gamma_{xz} = \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \varepsilon_{xx} & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{yx} & \varepsilon_{yy} & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{zx} & \frac{1}{2}\gamma_{zy} & \varepsilon_{zz} \end{bmatrix}$$

Deformation Model

- **Derivation of deformation for fracture**
  - Define a set of differential equations
  - Describe the aggregate behavior of the material in a continuous fashion
  - Use a finite element method to discretize these equations for computer simulation
  - Designed to be simple, fast, and suitable for fracture modeling

Continuous Model

- **Primary assumption**
  - The scale of the modeled effects is significantly greater than the scale of the material's composition
  - Macroscopic fractures can be significantly influenced by effects that occur at small scales
  - For graphical simulation, assume that a continuum model is adequate

Continuous Model

- **Define material coordinates**

$$\boldsymbol{u} = [u, v, w]^{\mathsf{T}}$$

- **Deformation of the material**

$$\boldsymbol{x}(\boldsymbol{u}) = [x, y, z]^{\mathsf{T}}$$

  - In areas where material exists, **x**(**u**) is continuous
  - Except across a finite number of surfaces within the volume that correspond to fractures

Continuous Model

- **Green's strain tensor**
  - Measure the local deformation of the material

$$\epsilon_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i} \cdot \frac{\partial \boldsymbol{x}}{\partial u_j} \right) - \delta_{ij}$$

    - Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & : \quad i = j \\ 0 & : \quad i \neq j \end{cases}$$

    - This strain metric only measures local deformation
    - Invariant with respect to rigid body transformations

Continuous Model

- **Strain rate tensor**
  - Measure the rate at which the strain is changing
  - Take the time derivative of strain

$$\nu_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i} \cdot \frac{\partial \dot{\boldsymbol{x}}}{\partial u_j} \right) + \left(\frac{\partial \dot{\boldsymbol{x}}}{\partial u_i} \cdot \frac{\partial \boldsymbol{x}}{\partial u_j} \right)$$

- Strain and strain rate tensors provide the raw information to compute internal elastic and damping forces
- But they do not take into account the properties of the material

Continuous Model

- **Stress tensor**
  - Represented as a 3 x 3 symmetric matrix
  - Two components
    - Elastic stress due to strain $\boldsymbol{\sigma}^{(\epsilon)}$
    - Viscous stress due to strain rate $\boldsymbol{\sigma}^{(\nu)}$
  - Total internal stress

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(\epsilon)} + \boldsymbol{\sigma}^{(\nu)}$$

  - Representation of elastic and viscous stresses

$$\sigma_{ij}^{(\epsilon)} = \sum_{k=1}^{3} \sum_{l=1}^{3} C_{ijkl}\, \epsilon_{kl}$$

$$\sigma_{ij}^{(\nu)} = \sum_{k=1}^{3} \sum_{l=1}^{3} D_{ijkl}\, \nu_{kl}$$

C: a set of the 81 elastic coefficients
D: a set of the 81 damping coefficients

Continuous Model

- **Stress tensor**
  - Coefficient reduction
    - Symmetric for both elastic and viscous stresses
    - Impose the additional constraint that the material is isotropic
    - **C** reduces to only two independent values (Lamé constants)

$$\sigma_{ij}^{(\epsilon)} = \sum_{k=1}^{3} \lambda \epsilon_{kk} \delta_{ij} + 2\mu\epsilon_{ij}$$

Material's rigidity : μ,
Resistance to changes in volume: λ

Continuous Model

- **Stress tensor**
  - Coefficient reduction
    - Similarly, **D** reduces to only two independent values

$$\sigma_{ij}^{(\nu)} = \sum_{k=1}^{3} \phi \nu_{kk} \delta_{ij} + 2\psi \nu_{ij}$$

    - The coefficients control how quickly the material dissipates internal kinetic energy
    - Will not damp motions that are locally rigid
    - Has the desirable property of dissipating only internal vibrations

Continuous Model

- **Potential densities**
  - Elastic potential density

$$\eta = \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \sigma_{ij}^{(\epsilon)} \epsilon_{ij}$$

  - Damping potential density

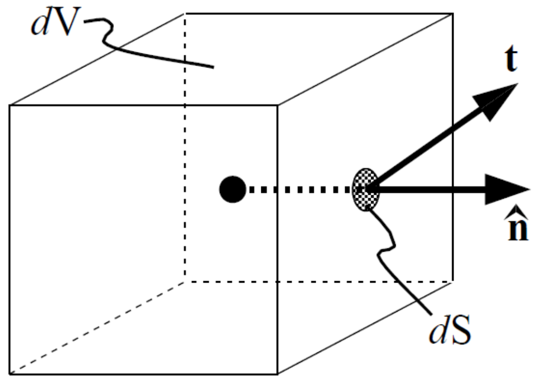
$$\kappa = \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \sigma_{ij}^{(\nu)} \nu_{ij}$$

  - Integrated over the volume of the material
    - Obtain the total elastic and damping potentials

Continuous Model

- **Internal force**
  - The stress can also be used to compute the forces acting internally to the material at any location
  - The traction (force per unit area) acting on a face perpendicular to the normal

$$\boldsymbol{t} = \boldsymbol{\sigma}\,\hat{\boldsymbol{n}}$$

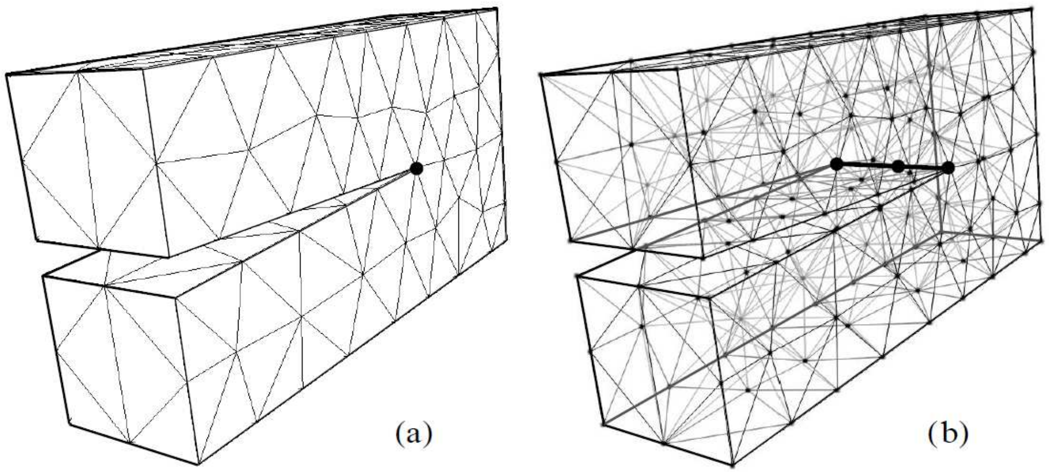


Finite Element Discretization

- **Discretization for computer simulation**
  - Finite difference
    - Well suited for problems with a regular structure but becomes complicated when the structure is irregular
  - Finite element
    - Partition the domain of the material into distinct sub-domains
    - Within each element, the material is described locally by a function
    - The function is decomposed into a set of shape (basis) functions, each associating one of the nodes
    - The mesh defines a piecewise function over the entire material domain

Finite Element Discretization

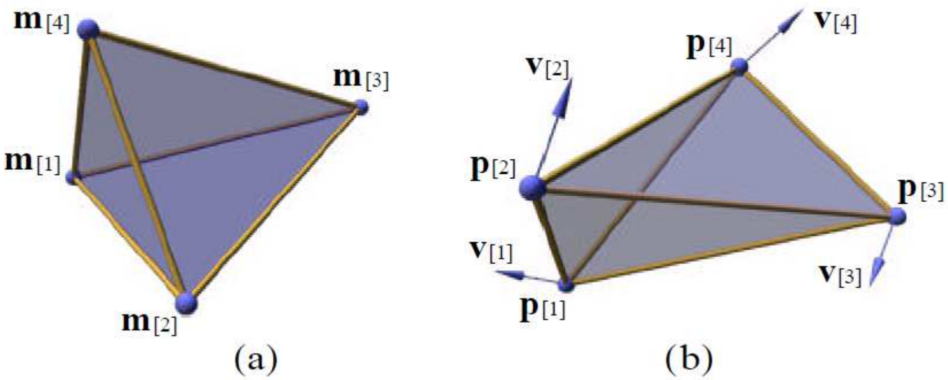
- **Tetrahedral mesh representation of objects**



(a) : only the external faces of the tetrahedra are drawn
(b) : the internal structure is shown

Finite Element Discretization

- **Formulation by barycentric coordinates**
  - Linear elements



- Each node has
  - A location in the material coordinate system (a)
  - A position and velocity in the world coordinate system (b)

Finite Element Discretization

- **Formulation by barycentric coordinates**
  - Linear elements
    - Barycentric coordinates provide a natural way to define the linear shape functions
    - Barycentric coordinates definition

$$\boldsymbol{b} = [b_1, b_2, b_3, b_4]^\mathsf{T}$$

    - Interpolate the entire region of the element

$$\begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{m}_{[1]} & \boldsymbol{m}_{[2]} & \boldsymbol{m}_{[3]} & \boldsymbol{m}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

Finite Element Discretization

- **Formulation by barycentric coordinates**
  - Linear elements
    - Barycentric coordinates may also be used to interpolate the node's world position and velocity

$$\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{[1]} & \boldsymbol{p}_{[2]} & \boldsymbol{p}_{[3]} & \boldsymbol{p}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{[1]} & \boldsymbol{v}_{[2]} & \boldsymbol{v}_{[3]} & \boldsymbol{v}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

    - Determine the barycentric coordinates of a point within the element

$$\boldsymbol{b} = \boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{m}_{[1]} & \boldsymbol{m}_{[2]} & \boldsymbol{m}_{[3]} & \boldsymbol{m}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1}$$

Finite Element Discretization

- **Formulation by barycentric coordinates**
  - Linear elements

$$\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{[1]} & \boldsymbol{p}_{[2]} & \boldsymbol{p}_{[3]} & \boldsymbol{p}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{[1]} & \boldsymbol{v}_{[2]} & \boldsymbol{v}_{[3]} & \boldsymbol{v}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

$$+ \qquad \boldsymbol{b} = \boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

$$\Downarrow$$

$$\boldsymbol{x}(\boldsymbol{u}) = \boldsymbol{P}\,\boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

$$\dot{\boldsymbol{x}}(\boldsymbol{u}) = \boldsymbol{V}\,\boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_{[1]} & \boldsymbol{p}_{[2]} & \boldsymbol{p}_{[3]} & \boldsymbol{p}_{[4]} \end{bmatrix}$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_{[1]} & \boldsymbol{v}_{[2]} & \boldsymbol{v}_{[3]} & \boldsymbol{v}_{[4]} \end{bmatrix}$$

Finite Element Discretization

- **Formulation by barycentric coordinates**
  - Linear elements
    - For non-degenerate elements, β is guaranteed to be non-singular
    - Computing the strain and strain rate tensors require

$$\frac{\partial \boldsymbol{x}}{\partial u_i} = \boldsymbol{P}\,\boldsymbol{\beta}\,\boldsymbol{\delta}_i \qquad \boldsymbol{\delta}_i = [\delta_{i1}\;\delta_{i2}\;\delta_{i3}\;0]^{\mathsf{T}}$$

$$\frac{\partial \dot{\boldsymbol{x}}}{\partial u_i} = \boldsymbol{V}\,\boldsymbol{\beta}\,\boldsymbol{\delta}_i$$

- Since the element's shape functions are linear, these partials are constant within the element

Finite Element Discretization

- **Computing forces at the node**
  - The element will exert elastic and damping forces on its nodes
    - Elastic force of the i-th node
      - Defined as the negative partial of the elastic potential density w.r.t. positions integrated over the volume of the element

$$\eta = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}^{(\epsilon)}\epsilon_{ij} \qquad \epsilon_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i}\cdot\frac{\partial \boldsymbol{x}}{\partial u_j}\right) - \delta_{ij} \qquad \frac{\partial \boldsymbol{x}}{\partial u_i} = \boldsymbol{P}\,\boldsymbol{\beta}\,\boldsymbol{\delta}_i$$

$$\boldsymbol{f}_{[i]}^{(\epsilon)} = -\frac{\mathrm{vol}}{2}\sum_{j=1}^{4}\boldsymbol{p}_{[j]}\sum_{k=1}^{3}\sum_{l=1}^{3}\beta_{jl}\beta_{ik}\sigma_{kl}^{(\epsilon)}$$

$$\mathrm{vol} = \frac{1}{6}[(\boldsymbol{m}_{[2]} - \boldsymbol{m}_{[1]}) \times (\boldsymbol{m}_{[3]} - \boldsymbol{m}_{[1]})] \cdot (\boldsymbol{m}_{[4]} - \boldsymbol{m}_{[1]})$$

Finite Element Discretization

- **Computing forces at the node**
  - The element will exert elastic and damping forces on its nodes
    - Damping force on the i-th node
      - Defined as the negative partial of the damping potential density w.r.t. velocity integrated over the volume of the element

$$\kappa = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}^{(\nu)}\nu_{ij} \qquad \nu_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i}\cdot\frac{\partial \dot{\boldsymbol{x}}}{\partial u_j}\right) + \left(\frac{\partial \dot{\boldsymbol{x}}}{\partial u_i}\cdot\frac{\partial \boldsymbol{x}}{\partial u_j}\right) \qquad \frac{\partial \dot{\boldsymbol{x}}}{\partial u_i} = \boldsymbol{V}\,\boldsymbol{\beta}\,\boldsymbol{\delta}_i$$

$$\boldsymbol{f}_{[i]}^{(\nu)} = -\frac{\mathrm{vol}}{2}\sum_{j=1}^{4}\boldsymbol{p}_{[j]}\sum_{k=1}^{3}\sum_{l=1}^{3}\beta_{jl}\beta_{ik}\sigma_{kl}^{(\nu)}$$

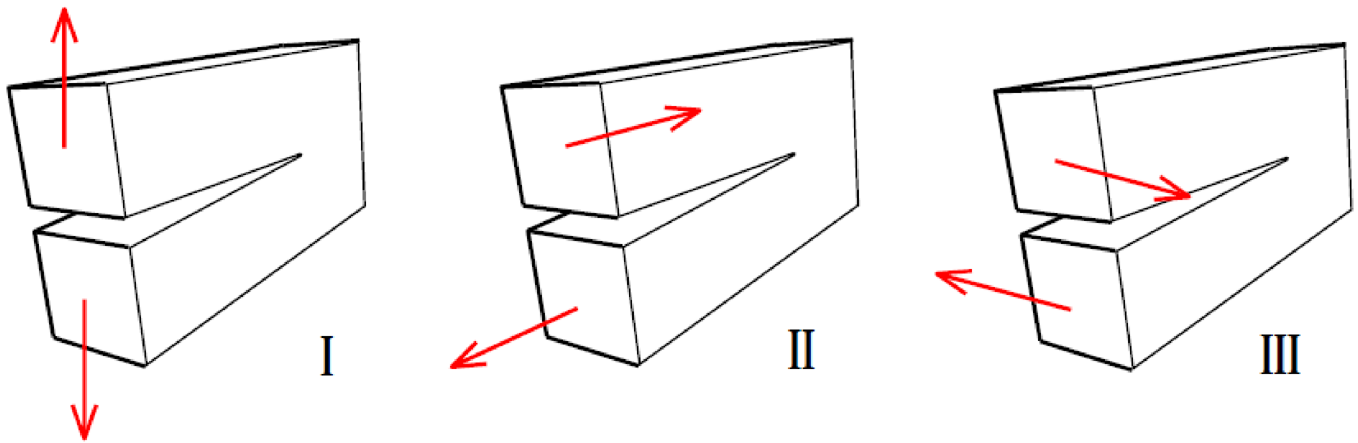
    - Total internal force that an element exerts on a node $\boldsymbol{f}_{[i]}^{\mathrm{el}} = -\frac{\mathrm{vol}}{2}\sum_{j=1}^{4}\boldsymbol{p}_{[j]}\sum_{k=1}^{3}\sum_{l=1}^{3}\beta_{jl}\beta_{ik}\sigma_{kl}$
    - Total force: summing all elements

Fracture Modeling

- **Based on linear elastic fracture mechanics**
  - The region of plasticity near the crack tip is neglected
  - Modeled materials will be brittle
    - Once the material has begun to fail, the fractures will have a strong tendency to propagate across the material
  - Three loading modes
    - Opening
    - In-plane shear
    - Out-of-plane shear

Fracture Modeling

- **Three loading modes experienced by a crack**



Mode I: Opening
Mode II: In-Plane Shear
Mode III: Out-of-Plane Shear

Fracture Modeling

- **Resolve the crack**
  - Analyze the forces acting at the crack tip
    - Tensile forces that are opposed by other tensile forces will cause the crack to continue
      - In a direction perpendicular to the direction of largest tensile load
    - Compressive loads will tend to arrest a crack
      - Perpendicular to the crack
    - Use the element nodes to determine where a crack should be initiated

Fracture Modeling

- **Fracture algorithm overview**
  - After each time step
    - Resolve the internal forces acting on all nodes
      - Tensile and compressive components
    - At each node
      - The forces are used to form a tensor
      - Describe how internal forces are acting to separate that node
      - If the action is sufficiently large, the node is split into two distinct nodes and a fracture plane is computed
      - All elements attached to the node are divided along the plane

Force Decomposition

- **Forces acting on a node**
  - Decomposed by separating the stress tensors into tensile and compressive components
  - Eigen analysis of stress tensor
    - Let $\mathrm{v}^i(\boldsymbol{\sigma}) \quad i \in \{1, 2, 3\}$ be the i-th eigenvalue of $\boldsymbol{\sigma}$
    - Let $\hat{\mathbf{n}}^i(\boldsymbol{\sigma})$ be the corresponding unit length eigenvector
    - Analysis
      - Positive eigenvalues correspond to tensile stresses
      - Negative eigenvalues to compressive stresses

Force Decomposition

- **Decomposition of stress tensor**
  - Given a vector **a** in R^3, we can construct a 3 x 3 symmetric matrix **m**(**a**)
    - |**a**| as an eigenvalue
    - **a** as the corresponding eigenvector
    - The other two eigenvalues equal to zero

$$\mathbf{m}(\boldsymbol{a}) = \begin{cases} \boldsymbol{a}\,\boldsymbol{a}^{\top}/|\boldsymbol{a}| & : \quad \boldsymbol{a} \neq \mathbf{0} \\ \mathbf{0} & : \quad \boldsymbol{a} = \mathbf{0} \end{cases}$$

Force Decomposition

- **Decomposition of stress tensor**
  - The tensile and compressive components can be represented as

$$\boldsymbol{\sigma}^{+} = \sum_{i=1}^{3} \max(0, \mathrm{v}^{i}(\boldsymbol{\sigma}))\, \mathbf{m}(\hat{\mathbf{n}}^{i}(\boldsymbol{\sigma}))$$

$$\boldsymbol{\sigma}^{-} = \sum_{i=1}^{3} \min(0, \mathrm{v}^{i}(\boldsymbol{\sigma}))\, \mathbf{m}(\hat{\mathbf{n}}^{i}(\boldsymbol{\sigma}))$$

  - Force that an element exerts on a node
    - Tensile component $\boldsymbol{f}^{+}_{[i]} = -\frac{\mathrm{vol}}{2} \sum_{j=1}^{4} \boldsymbol{p}_{[j]} \sum_{k=1}^{3} \sum_{l=1}^{3} \beta_{jl} \beta_{ik} \sigma^{+}_{kl}$
    - Compressive component using $\boldsymbol{f}_{[i]} = \boldsymbol{f}^{+}_{[i]} + \boldsymbol{f}^{-}_{[i]}$

Separation Tensor

- **Separation tensor**
  - Used directly to determine whether a fracture should occur at a node
  - Formed from the balanced tensile and compressive forces acting at each node

$$\varsigma = \frac{1}{2}\left(-\mathbf{m}(\boldsymbol{f}^{+}) + \sum_{\boldsymbol{f}\in\{\boldsymbol{f}^{+}\}} \mathbf{m}(\boldsymbol{f}) + \mathbf{m}(\boldsymbol{f}^{-}) - \sum_{\boldsymbol{f}\in\{\boldsymbol{f}^{-}\}} \mathbf{m}(\boldsymbol{f})\right)$$

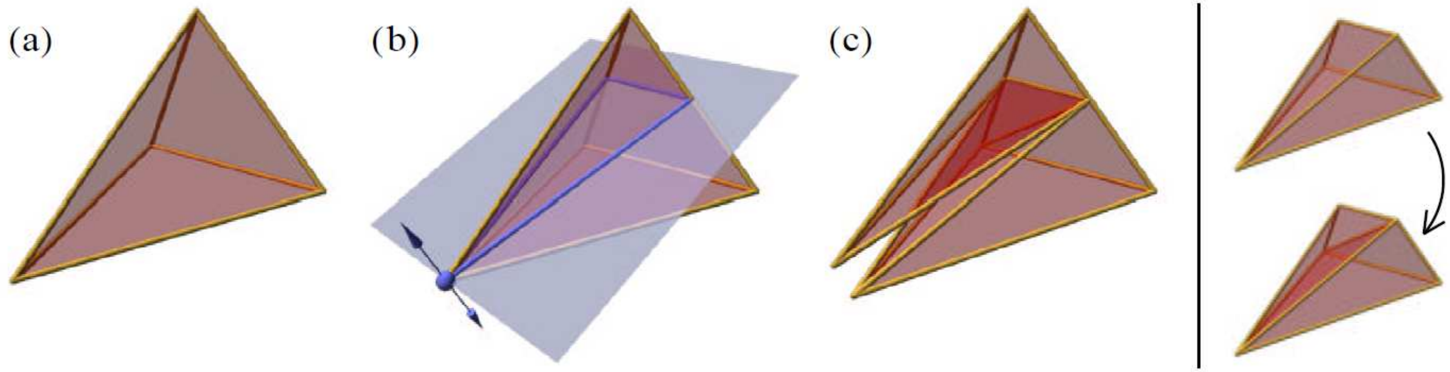
  - Let v^{+} be the largest positive eigenvalue, If v^{+} is greater than the material toughness τ, the material will fail at the node

Fracture Plane

- **Orientation of the fracture plane**
  - Perpendicular to the tensile eigen vector
  - In case of multiple eigenvalues
    - Multiple fracture planes may be generated
    - First generate the plane for the largest value
    - Remeshing
    - Recompute the new value for separation tensor and proceed the above

Fracture Plane

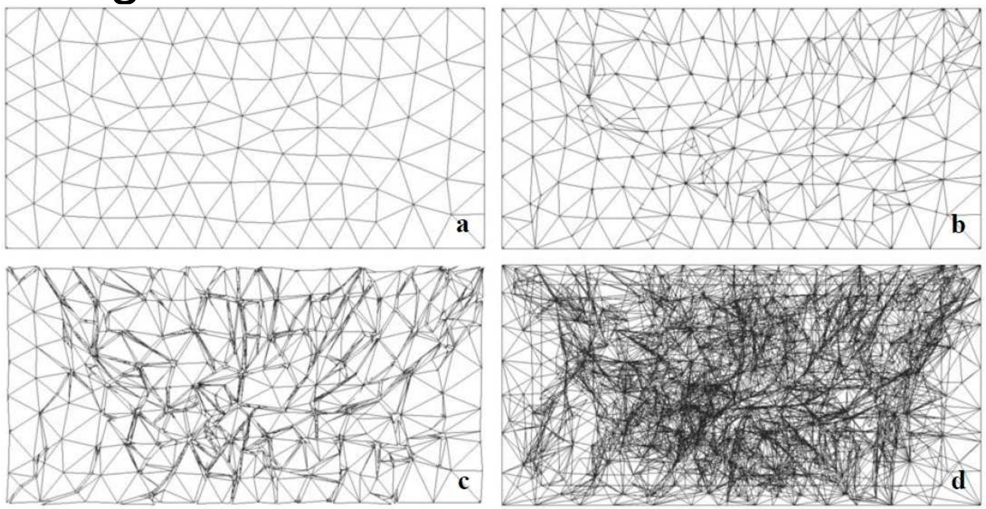
- **An element is split by the fracture plane**



(a) The initial tetrahedral element
(b) The splitting node and fracture plane are shown in blue
(c) The element is split along the fracture plane into two polyhedra

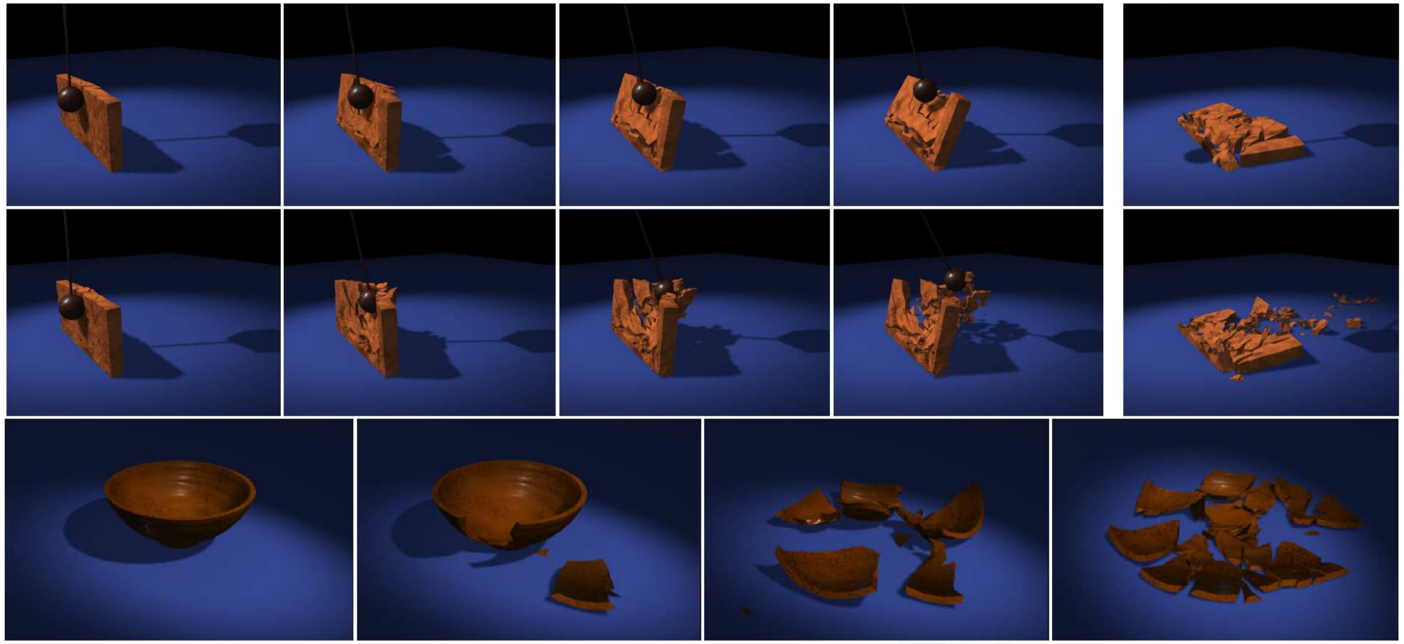
Local Remeshing

- **The mesh must be modified to reflect the new discontinuity**
  - Split elements that intersect the fracture plane
  - Modify neighboring elements to ensure mesh consistency



Simulation Results

- **Fracture-based rigid body simulation**

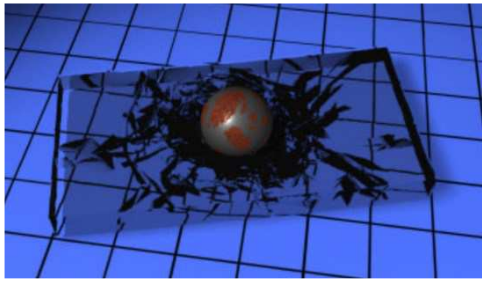


Simulation Results

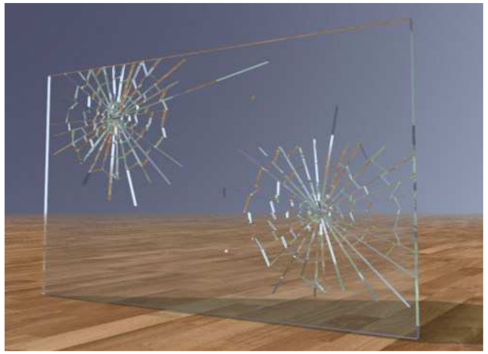
- **Fracture-based rigid body simulation**

Fracture Animation

- **Physically simulated fracture**
  - Physically accurate
  - Stability issue
  - Slow in high resolution



- **Pre-defined fracture pattern**
  - Easier artistic control
  - Fast and robust
  - Difficult to create physically plausible details

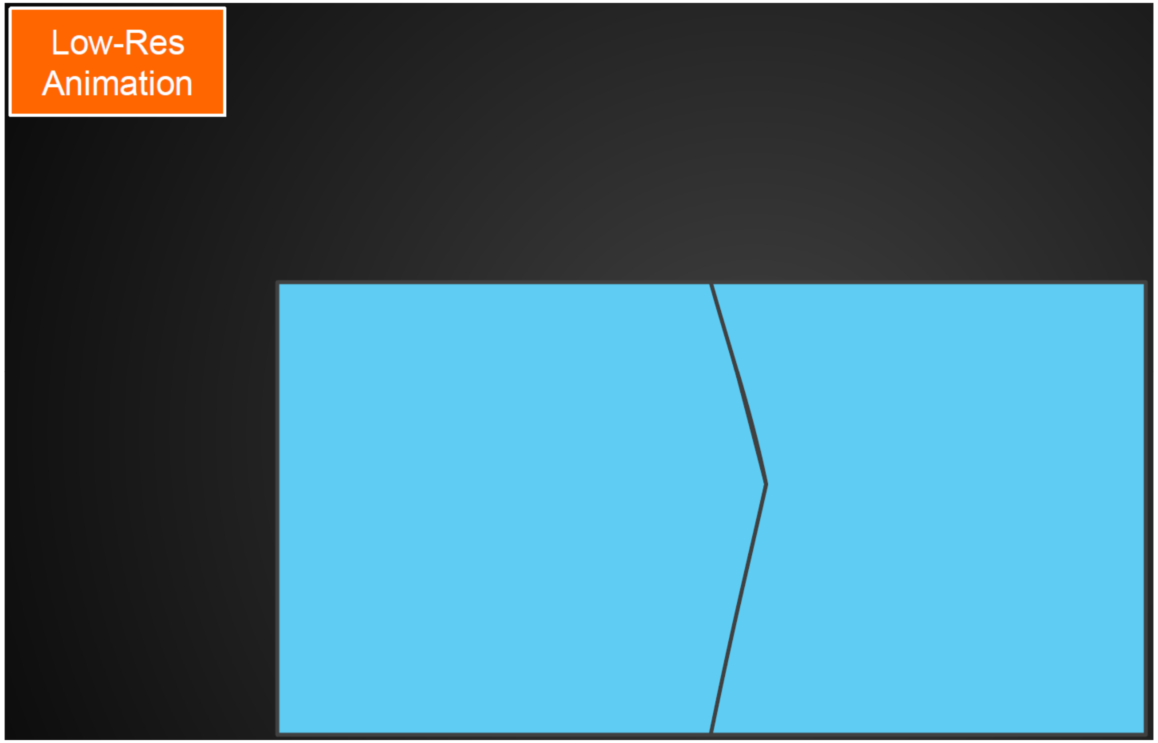


Adaptive Fracture Refinement

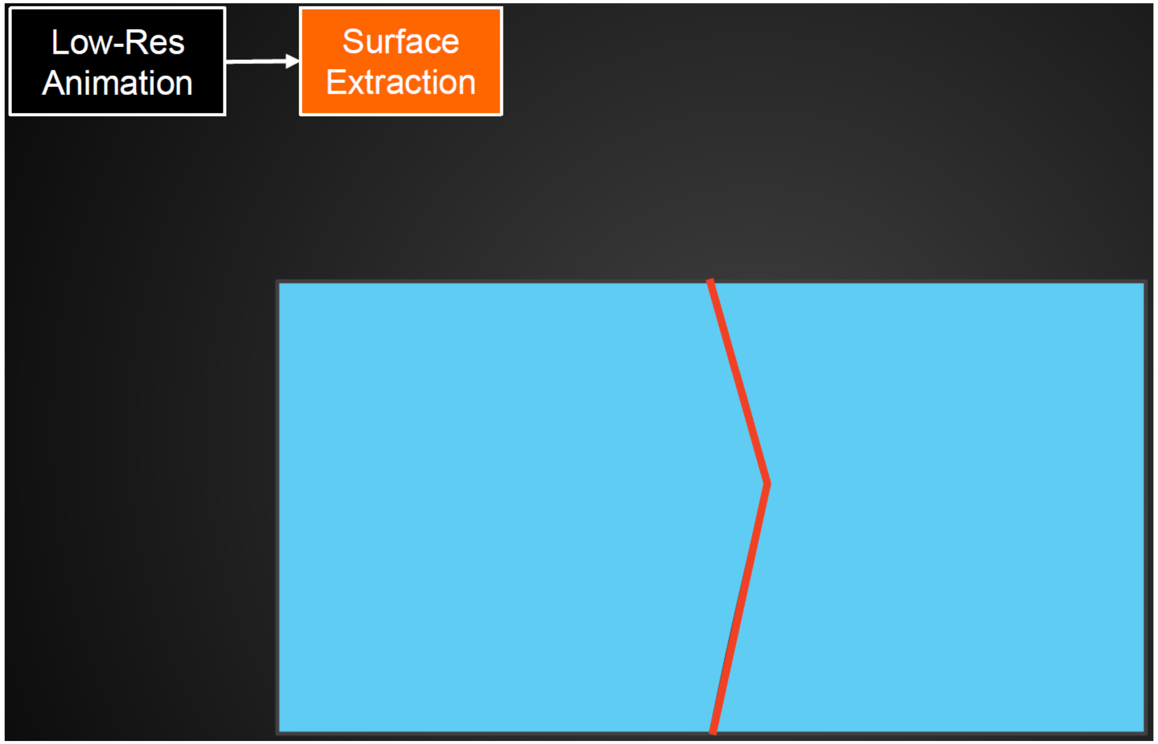
- **Fracture is mesh resolution dependent**
  - Low resolution mesh simulation can be dramatically different from high resolution ones
  - High resolution simulation
    - Collision detection is computationally costly
    - Fracture computation time is high

- **Adaptive refinement**
  - Given a low resolution fracture animation
    - Adaptively refine the fracture surface within a 3D object

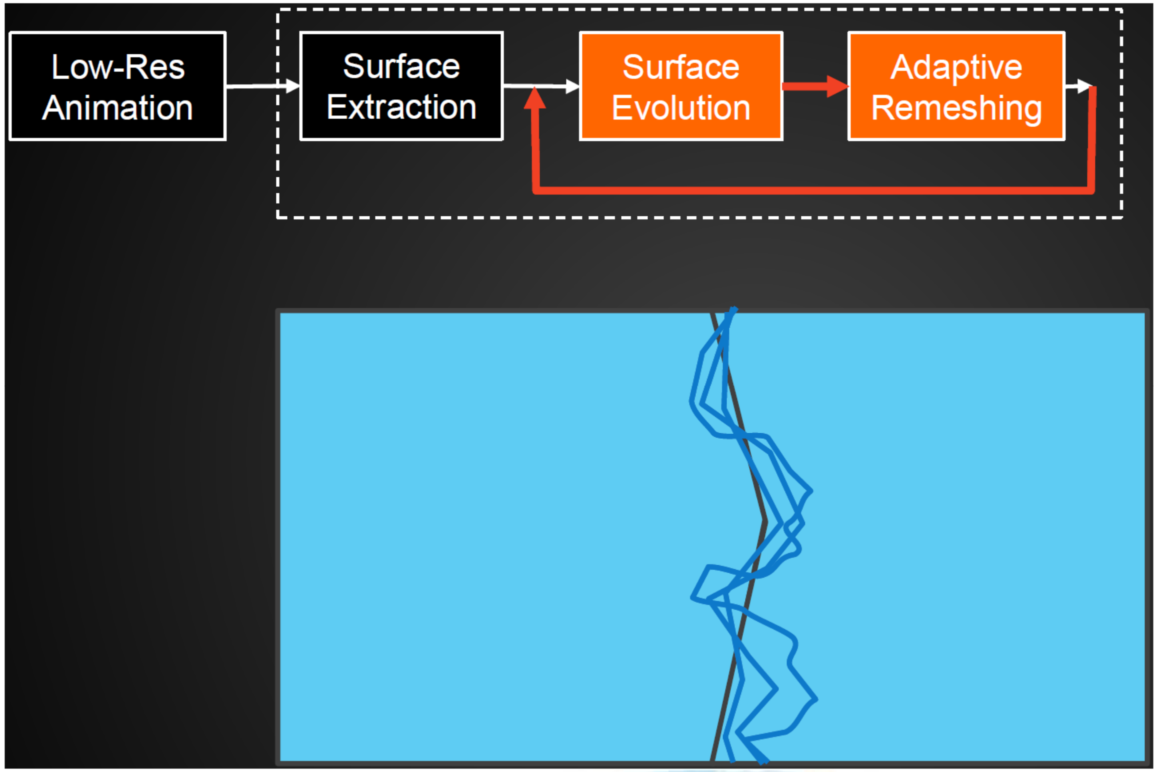
Adaptive Fracture Refinement



Adaptive Fracture Refinement

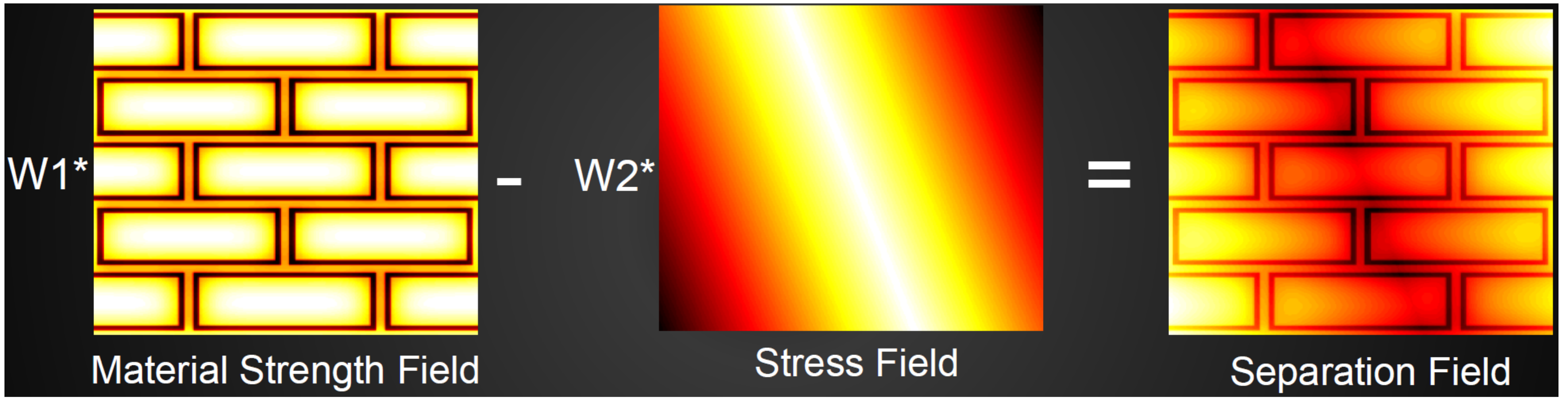


Adaptive Fracture Refinement



Fracture Surface Evolution

- **Separation field**
  - Where the stress is high and material strength is weak
  - Vertices should move to the lowest value regions



Gradient Flow & Surface Evolution

- **Evolve surface S to minimize**

modified strength field

$$\mathcal{E}(\mathbf{S}) = \int_{\mathbf{S}} \psi(x)\,ds$$

  - Gradient descent for each vertex

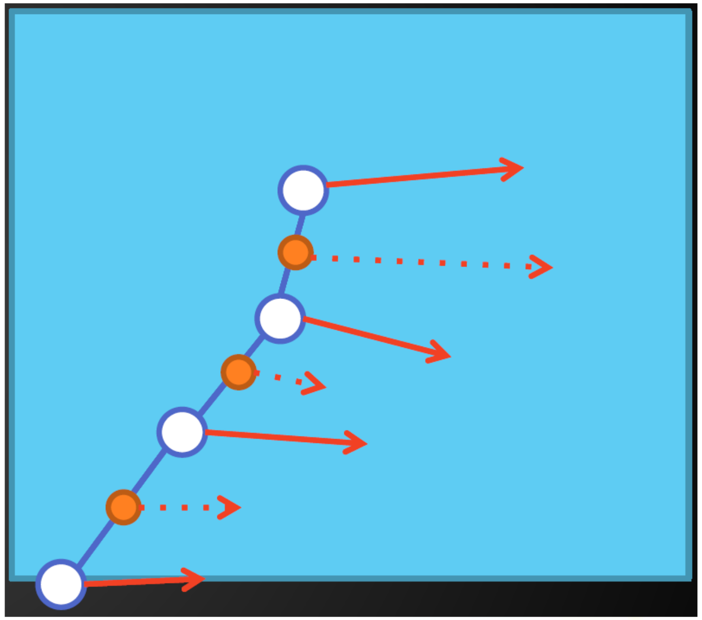
$$\frac{d\mathbf{x}_i}{dt} = -\frac{1}{A_i} \sum_{j \in N_i} \left(\int_{\mathbf{S}_j} \nabla_{\mathbf{x}_i} \psi(\mathbf{x}) \phi_i(\mathbf{x})\,ds - \frac{\mathbf{e}_j^i \times \mathbf{n}_j}{2A_j} \int_{\mathbf{S}_j} \psi(\mathbf{x})\,ds \right)$$

  - Approximation
    - Separation field varies linearly within triangle plane

$$\frac{d\mathbf{x}_i}{dt} = -\frac{1}{A_i} \sum_{j \in N_i} \left(\frac{1}{3} A_j \nabla \psi(\mathbf{x}_i) - \frac{\mathbf{e}_j^i \times \mathbf{n}_j}{2A_j} \sum_{k \in T_j} \psi(\mathbf{x}_k) \right)$$

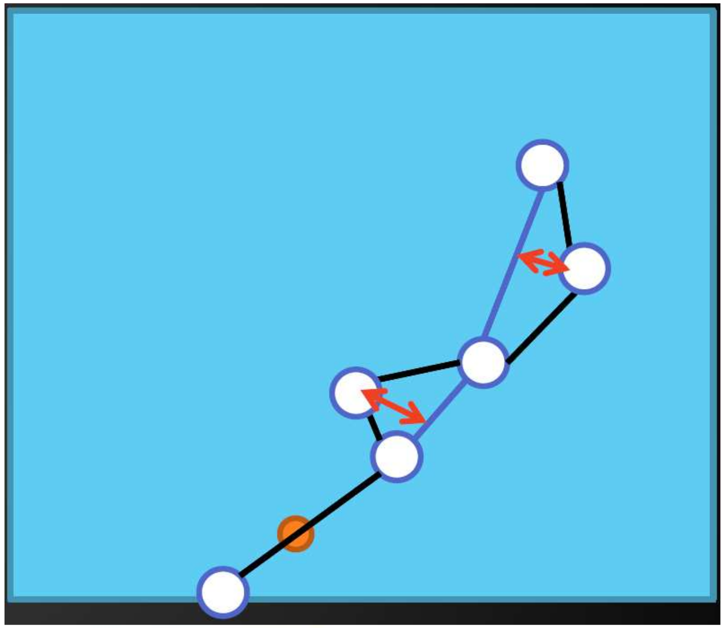
Adaptive Fracture Remeshing

- **Random candidate vertices**



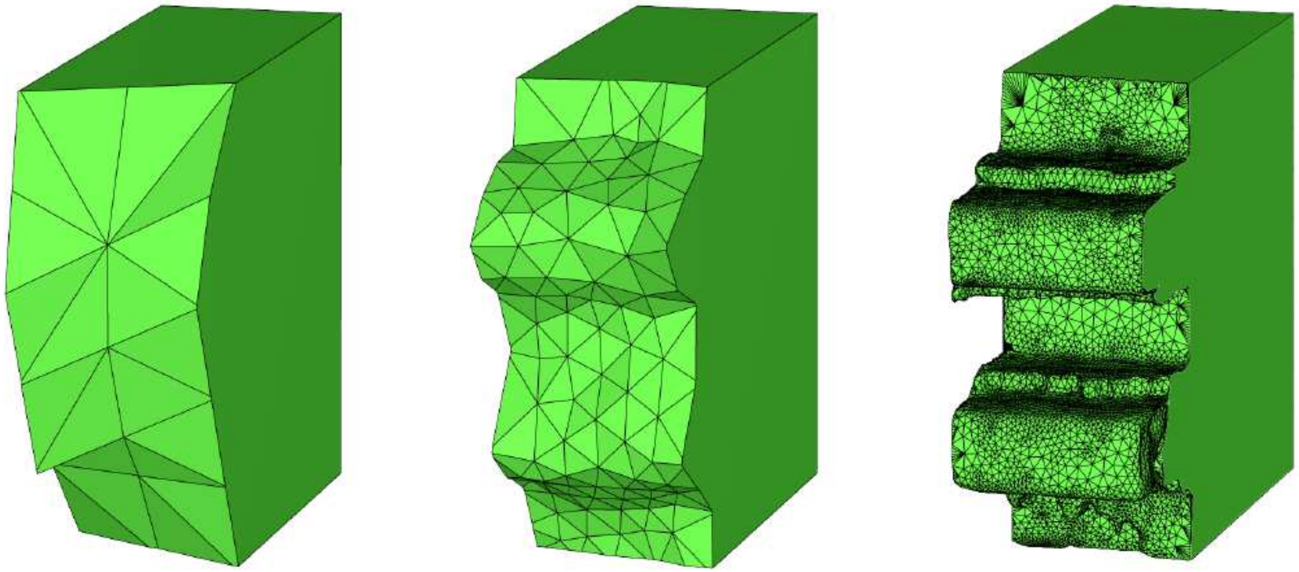
Adaptive Fracture Remeshing

- **Random candidate vertices**
  - Select and insert candidates
  - Edge flipping optimization



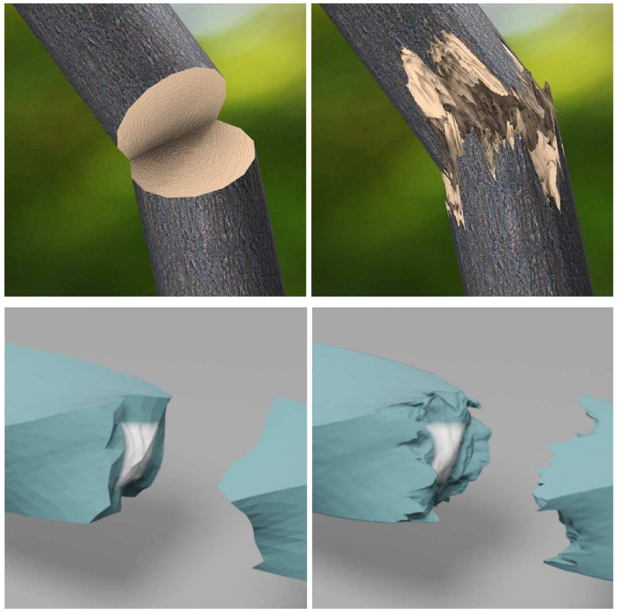
Adaptive Fracture Remeshing

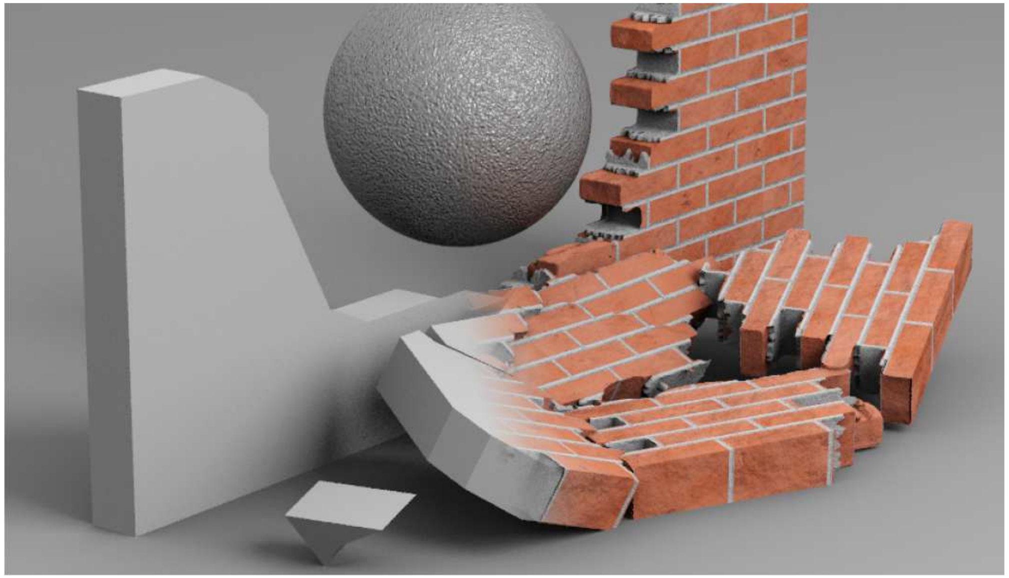
- **Iterative remeshing**



Results

- **Fine-detail fractures**





Results

Physics-Inspired Adaptive Fracture Refinement

Zhili Chen   Miaojun Yao   Renguo Feng   Huamin Wang

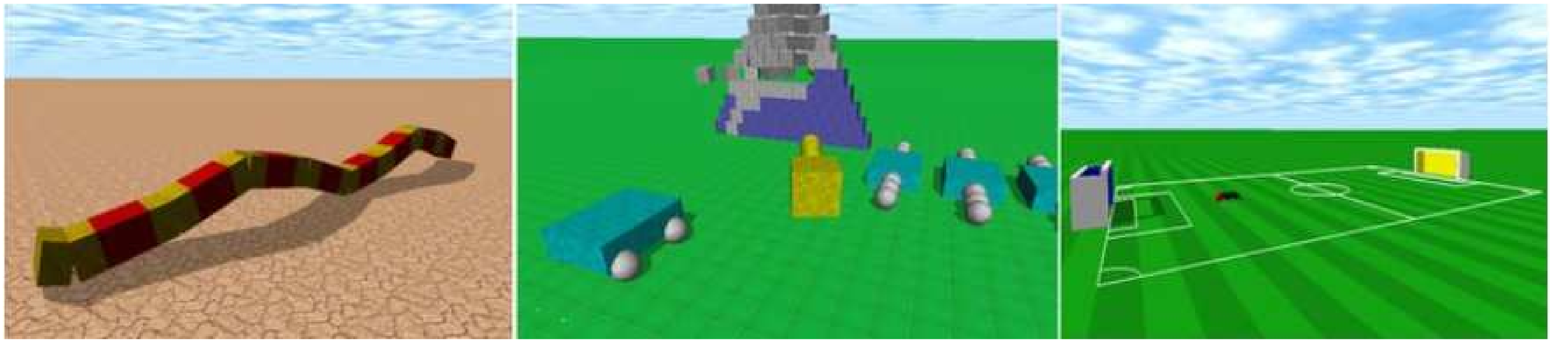
The Ohio State University

SIGGRAPH 2014

Advanced Real-Time Rigid Body Fracture

Rigid Body Simulation Software

- **Open Dynamics Engine (ODE)**
  - An open source, high performance library for simulating rigid body dynamics
  - https://www.ode.org/



Rigid Body Simulation Software

- **Bullet Physics Engine**
  - Real-time collision detection and multi-physics simulation for VR, games, visual effects, robotics, machine learning etc
  - https://pybullet.org/wordpress/
  - https://github.com/bulletphysics/bullet3

Next Lecture : Soft-Body Simulation – Hair I