



Imagine Day 2009

破壊シミュレーション入門

株式会社バンダイナムコゲームス コンテンツ制作本部 技術部 プロジェクトサポート課
ソフトウェアテクノロジスト 今給黎 隆

- 🍵 はじめに
- 🍵 歴史
- 🍵 破壊シミュレーション
- 🍵 細かな破壊の表現
- 🍵 まとめ

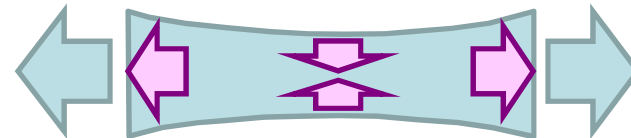
- 🍵 はじめに
- 🍵 歴史
- 🍵 破壊シミュレーション
- 🍵 細かな破壊の表現
- 🍵 まとめ

- 🍵 本講演では数式を多用しております
- 🍵 本講演ではシェーダの話はできません
 - 🍵 社内で予行演習を行った際にGPUの話をしていいたら「懐かしい香りがする」言われたので、方向性を90度変えています

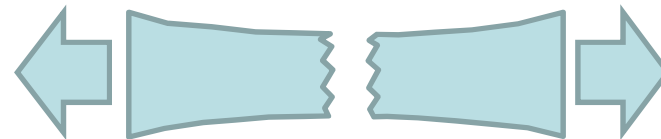
物を引っ張ると



内部で力が発生して



限界を超えると切れる



🍵 商用ライブラリの発展

🍵 Havok Destruction

🍵 NVIDIA APEX Destruction

🍵 破壊に重点をおいたゲームの出現

🍵 レッドファクション:ゲリラ

🍵 人為的

- 🍵 **車両の衝突...** 道路の分離帯、路側帯のガードフェンスなど (薄い物体は対象外)
- 🍵 **重量物の落下...** 工場、建設現場など
- 🍵 **高所からの落下...** 高層マンション、高層ビルなど
- 🍵 **航空機の落下事故...** (構造が複雑な物体は対象外)
- 🍵 **船舶の衝突事故...**
- 🍵 **テロによる攻撃...**
- 🍵 **ミサイル砲弾...**

🍵 自然的

- 🍵 **落石、岩盤崩壊...** 落石覆行など
- 🍵 **土石流...** 砂防ダムなど (細かな物体の運動は正確ではない)
- 🍵 **崩壊土砂...** 家屋など
- 🍵 **波浪、津波...** 防波堤、護岸構造物、港湾施設など (波は対象外)
- 🍵 **直下地震...** 橋脚、建物など
- 🍵 **竜巻...** 家屋、ビル破壊など
- 🍵 **溶解...** (徐々に融ける現象は対象外)

🍵はじめに

🍵歴史

🍵破壊シミュレーション

🍵細かな破壊の表現

🍵まとめ

Modeling inelastic deformation: viscoelasticity, plasticity, fracture

(非弾性変形のモデリング: 粘弾性、塑性、破壊)

Demetri Terzopoulos, Kurt W. Fleischer (SIGGRAPH 1988)

紙のような薄い膜をぴりぴりと破く

Animation of fracture by physical modeling

(物理モデルに基づく破砕アニメーション)

Alan Norton, Greg Turk, Bob Bacon, John Gerth, Paula Sweeney (TVC 1991)

初3D破壊

注：読み難い

Fast and controllable simulation of the shattering of brittle objects

(脆い物体の粉砕に関する高速で制御可能なアニメーション)

Jeffrey Smith, Andrew Witkin and David Baraff (Graphics Interface 2000)

高速化

Graphical modeling and animation of brittle fracture

(物理モデルに基づく破砕アニメーション)

James F. O'Brien and Jessica K. Hodgins (SIGGRAPH '99)

Real-time simulation of deformation and fracture of stiff materials

(堅い物体における変形と破砕のリアルタイムシミュレーション)


Matthias Muller, Leonard McMillan, Julie Dorsey and Robert Jagnow (CAS 2001)

Physically-based simulation of objects represented by surface meshes

(表面メッシュで表現された物体の物理ベースシミュレーション)

Matthias Muller, Matthias Teschner, and Markus Gross (CGI2004)

メッシュレスベース手法

 物体に粒子を振りまいて、
粒子単位で変形、分裂を計算

形状マッチング

 ブロック等を繋げて形状変形を計算

🍵 カリフォルニア大学バークレー校 准教授

🍵 物理ベースの複雑な変形と流体

🍵 Graphical modeling and animation of brittle fracture

(物理モデルに基づく破砕アニメーション)

James F. O'Brien and Jessica K. Hodgins (SIGGRAPH '99)

🍵 GDC2009 : Real-Time Deformation and Fracture: Finite element simulation and its use in "Star Wars The Force Unleashed"

🍷 Real-Time Deformation and Fracture in a Game Environment (SCA2009)

🍷 事前分割した物体を用意

🍷 有限要素法で破断を計算

🍷 分割した物体の中心が含まれる

要素の位置、向きの情報から

各物体の位置を決定

Research Lead PhysX SDK NVIDIA

Real-time simulation of deformation and fracture of stiff materials

(堅い物体における変形と破砕のリアルタイムシミュレーション)

Matthias Muller, Leonard McMillan, Julie Dorsey and Robert Jagnow (CAS 2001)

Physically-based simulation of objects represented by surface meshes

(表面メッシュで表現された物体の物理ベースシミュレーション)

Matthias Muller, Matthias Teschner, and Markus Gross (CGI2004)

HPは物理エンジンを作る人にお勧め

- 🍵 はじめに
- 🍵 歴史
- 🍵 **破壊シミュレーション**
- 🍵 細かな破壊の表現
- 🍵 まとめ

🍵 マテリアル座標系: $\mathbf{u} = (u_0, u_1, u_2) = (u, v, w)$

🍵 モデリングツールで作った時の頂点(や補間された点)の位置

🍵 ワールド座標系: $\mathbf{x} = (x_0, x_1, x_2) = (x, y, z)$

🍵 物体の実際の空間での位置

🍵 マテリアル空間で長さ Δu

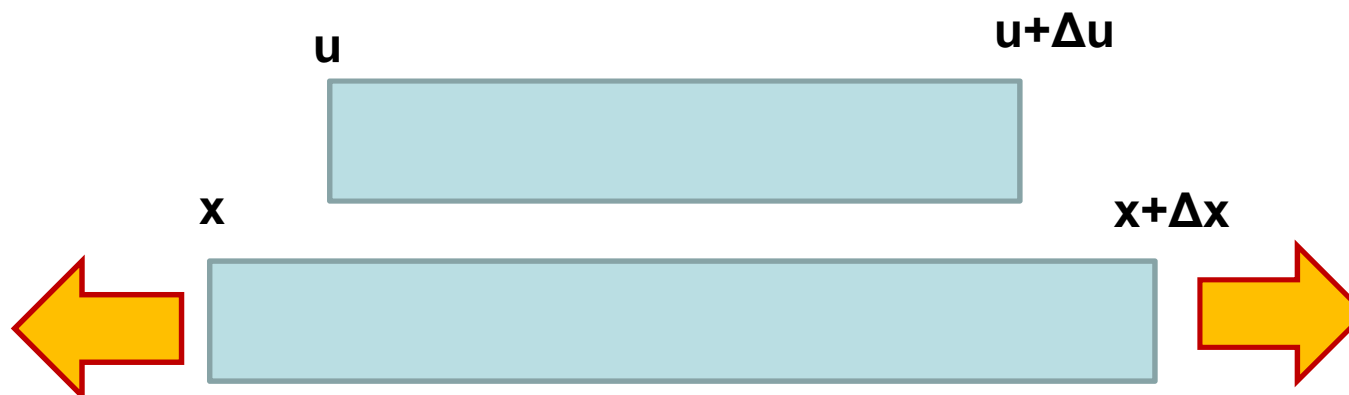
🍵 ワールド空間で Δx (引き延ばされた)

🍵 引き延ばされた割合

$$\frac{\Delta x}{\Delta u} \rightarrow \frac{dx}{du} \quad \text{小さな一部分に注目したときの伸び率}$$

$$\rightarrow \frac{\partial \mathbf{x}}{\partial u_i} \quad \text{3次元の場合のある方向について}$$

$$u_0 = u, u_1 = v, u_2 = w$$



🍵 変形のものさし

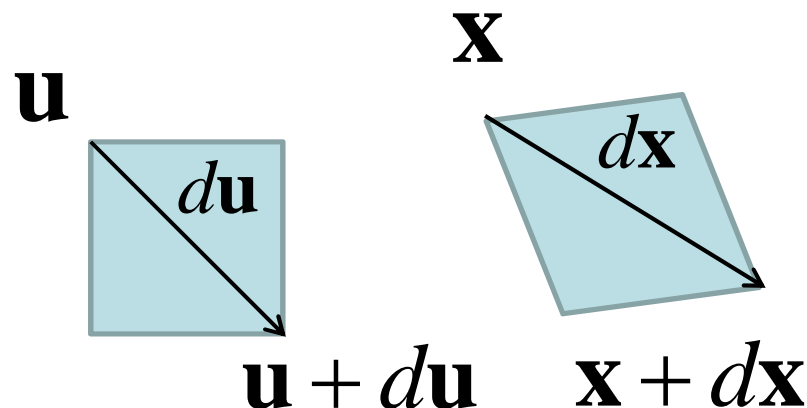
🍵 物体の一部が
変形した時の関係

$$dx_i = \frac{\partial x_i}{\partial u_j} du_j$$

$$dx = \frac{\partial x}{\partial u} du + \frac{\partial x}{\partial v} dv + \frac{\partial x}{\partial w} dw$$

$$dy = \frac{\partial y}{\partial u} du + \frac{\partial y}{\partial v} dv + \frac{\partial y}{\partial w} dw$$

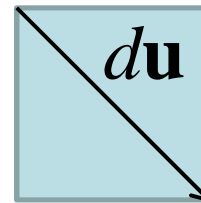
$$dz = \frac{\partial z}{\partial u} du + \frac{\partial z}{\partial v} dv + \frac{\partial z}{\partial w} dw$$



🍵 変位を対称と非対称に分解

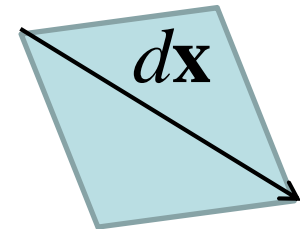
$$dx_i = \frac{\partial x_i}{\partial u_j} du_j \iff \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial w} \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix}$$

u



u + du

x



x + dx

$$\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \frac{\partial x}{\partial u} + \frac{\partial x}{\partial w} & \frac{\partial x}{\partial v} & \frac{\partial x}{\partial u} - \frac{\partial x}{\partial w} \\ \frac{\partial y}{\partial u} + \frac{\partial y}{\partial w} & \frac{\partial y}{\partial v} & \frac{\partial y}{\partial u} - \frac{\partial y}{\partial w} \\ \frac{\partial z}{\partial u} + \frac{\partial z}{\partial w} & \frac{\partial z}{\partial v} & \frac{\partial z}{\partial u} - \frac{\partial z}{\partial w} \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 & -(\text{rot}\mathbf{x})_z & (\text{rot}\mathbf{x})_y \\ (\text{rot}\mathbf{x})_z & 0 & -(\text{rot}\mathbf{x})_x \\ -(\text{rot}\mathbf{x})_y & (\text{rot}\mathbf{x})_x & 0 \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix}$$

歪み

回転

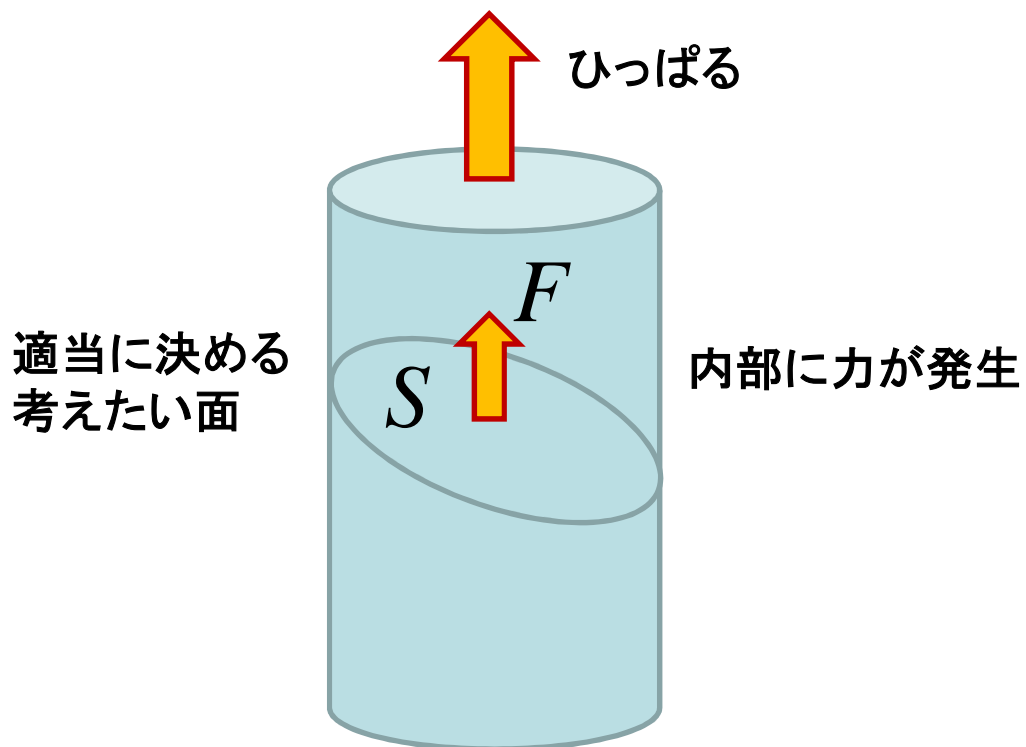
$$= \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{1}{2} \left(\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u} \right) & \frac{1}{2} \left(\frac{\partial x}{\partial w} + \frac{\partial z}{\partial u} \right) \\ \frac{1}{2} \left(\frac{\partial y}{\partial u} + \frac{\partial x}{\partial v} \right) & \frac{\partial y}{\partial v} & \frac{1}{2} \left(\frac{\partial y}{\partial w} + \frac{\partial z}{\partial v} \right) \\ \frac{1}{2} \left(\frac{\partial z}{\partial u} + \frac{\partial x}{\partial w} \right) & \frac{1}{2} \left(\frac{\partial z}{\partial v} + \frac{\partial y}{\partial w} \right) & \frac{\partial z}{\partial w} \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix}$$

$$+ \frac{1}{2} \begin{pmatrix} 0 & -(\text{rot}\mathbf{x})_z & (\text{rot}\mathbf{x})_y \\ (\text{rot}\mathbf{x})_z & 0 & -(\text{rot}\mathbf{x})_x \\ -(\text{rot}\mathbf{x})_y & (\text{rot}\mathbf{x})_x & 0 \end{pmatrix} \begin{pmatrix} du \\ dv \\ dw \end{pmatrix}$$

コーシーの歪みテンソル

$$\text{rot}\mathbf{x} = \nabla_{\mathbf{u}} \times \mathbf{x} = \begin{pmatrix} \frac{\partial z}{\partial v} - \frac{\partial y}{\partial w} & \frac{\partial x}{\partial w} - \frac{\partial z}{\partial u} & \frac{\partial y}{\partial u} - \frac{\partial x}{\partial v} \end{pmatrix}$$

🍵 応力：単位面積あたりの力

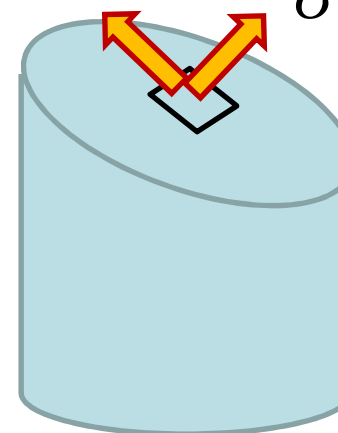


ある面の接線の方
向へのすべりの力
(せん断応力)

$$\tau = \frac{\mathbf{F} - (\mathbf{F} \cdot \mathbf{n})\mathbf{n}}{S}$$

ある面の法線の方
向にかかる力
(垂直応力)

$$\sigma = \frac{\mathbf{F} \cdot \mathbf{n}}{S}$$



🍵 ばねはある範囲なら掛かる力はばねの長さに比例する

🍵 だからばねは、はかりになる

$$f = -k x$$

ばね定数

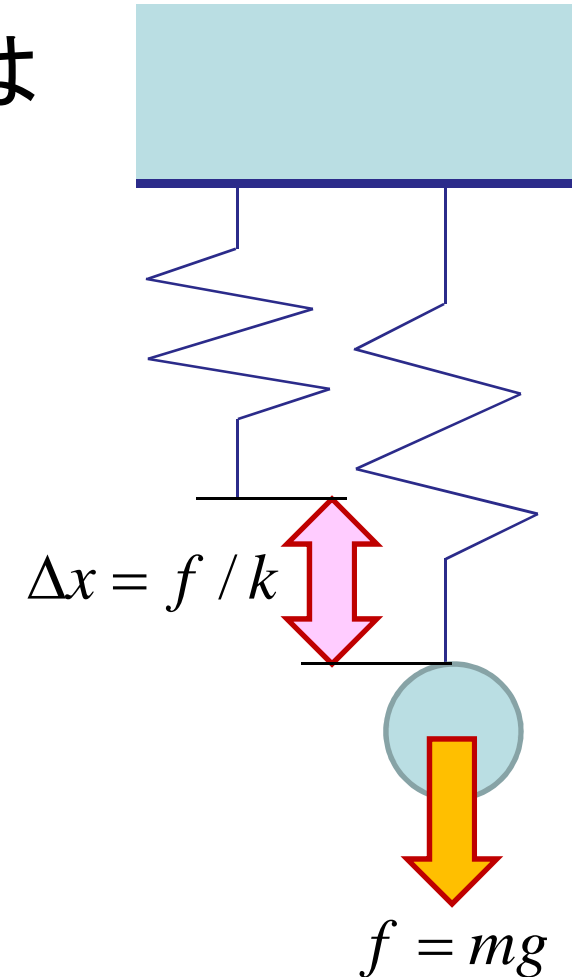
🍵 弾性体(変形する物体)の場合

$$\sigma_{ij}^{(\varepsilon)} = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl} \varepsilon_{kl}$$

応力

歪み

係数(3x3x3x3=81個のパラメータ)



🍵 材質が等方的な場合

🍵 木の内部の様な特別な方向性を持たない

$$\sigma_{ij}^{(\varepsilon)} = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl} \varepsilon_{kl}$$

🍵 座標系のある成分の符号を変えても値は同じ

例えば: $x \rightarrow -x$

➡ C_{0001} のような、同じ添え字が来るような成分は0になる

🍵 座標系を入れ替えても値は変わらない

➡ $C_{0011} = C_{1100}$ のように、添え字を入れ替えた場合の値は同じ

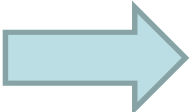
$$C_{ijkl} = A\delta_{ij}\delta_{kl} + B\delta_{ik}\delta_{lj} + C\delta_{il}\delta_{jk}$$

Lame の定数 (2/2)



$$\sigma_{ij}^{(\varepsilon)} = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl} \varepsilon_{kl}$$

$$C_{ijkl} = A\delta_{ij}\delta_{kl} + B\delta_{ik}\delta_{lj} + C\delta_{il}\delta_{jk}$$


$$\sigma_{ij}^{(\varepsilon)} = \sum_{k=0}^2 \sum_{l=0}^2 (A\delta_{ij}\delta_{kl} + B\delta_{ik}\delta_{lj} + C\delta_{il}\delta_{jk}) \varepsilon_{kl}$$

$$= A\delta_{ij}\varepsilon_{kk} + B\varepsilon_{ij} + C\varepsilon_{ji}$$

$$\sigma_{ij}^{(\varepsilon)} = \lambda(\text{Tr } \varepsilon_{ij})\delta_{ij} + 2\mu\varepsilon_{ij}$$

ラメの弾性定数

こんな計算して何がうれしいの？



🍵 実測できる値との対応

🍵 単位面積当たりfの力で引っ張る

$$f = F / S$$



$$\sigma_{ij}^{(\varepsilon)} = \lambda (\text{Tr } \varepsilon_{ij}) \delta_{ij} + 2\mu \varepsilon_{ij}$$

$$\sigma_{00}^{(\varepsilon)} = \lambda (\text{Tr } \varepsilon) \delta_{00} + 2\mu \varepsilon_{00} = f$$

$$\sigma_{11}^{(\varepsilon)} = \lambda (\text{Tr } \varepsilon) \delta_{11} + 2\mu \varepsilon_{11} = 0 \quad \sigma_{01}^{(\varepsilon)} = \sigma_{12}^{(\varepsilon)} = \sigma_{20}^{(\varepsilon)} = 0$$

$$\sigma_{22}^{(\varepsilon)} = \lambda (\text{Tr } \varepsilon) \delta_{22} + 2\mu \varepsilon_{22} = 0$$

➡ $\lambda = \frac{E}{(1-2\sigma)(1+\sigma)}, \quad \mu = \frac{E}{2(1+\sigma)}$

$E = f / \varepsilon_{00}$: ヤング率 (一方向の引っ張り(圧縮)応力の方向に対する歪み量の関係)

$\sigma = -\varepsilon_{11} / \varepsilon_{00}$: ポアソン比 (軸方向とそれに直交する方向の歪みとの比)

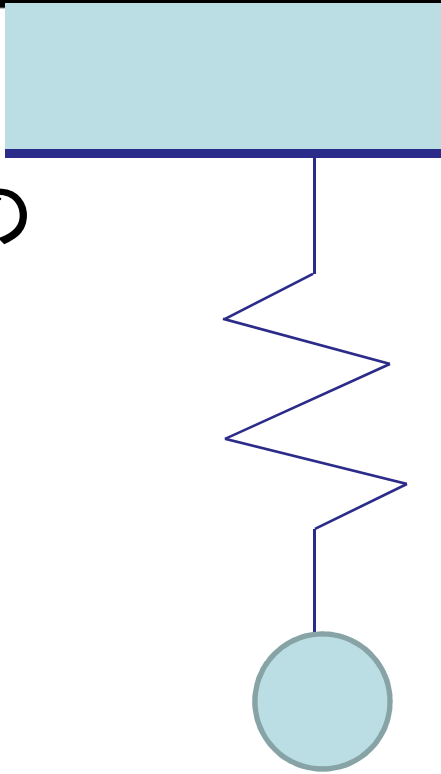
🍵 エネルギー:

🍵 時間がたっても総和が変わらないもの

$$E = \frac{1}{2} m \mathbf{v}^2 + \frac{1}{2} k \mathbf{x}^2 - mgh$$

🍵 弾性体(変形する物体)の場合

$$U = \frac{1}{2} \sum_{i=0}^2 \sum_{j=0}^2 \sigma_{ij}^{(\varepsilon)} \varepsilon_{ij} = \frac{1}{2} \boldsymbol{\sigma}^{(\varepsilon)} : \boldsymbol{\varepsilon}$$


$$f = -k x$$

🍵 微小な変位が与えられた時、物体全体の応力と外力の仮想仕事は等しい

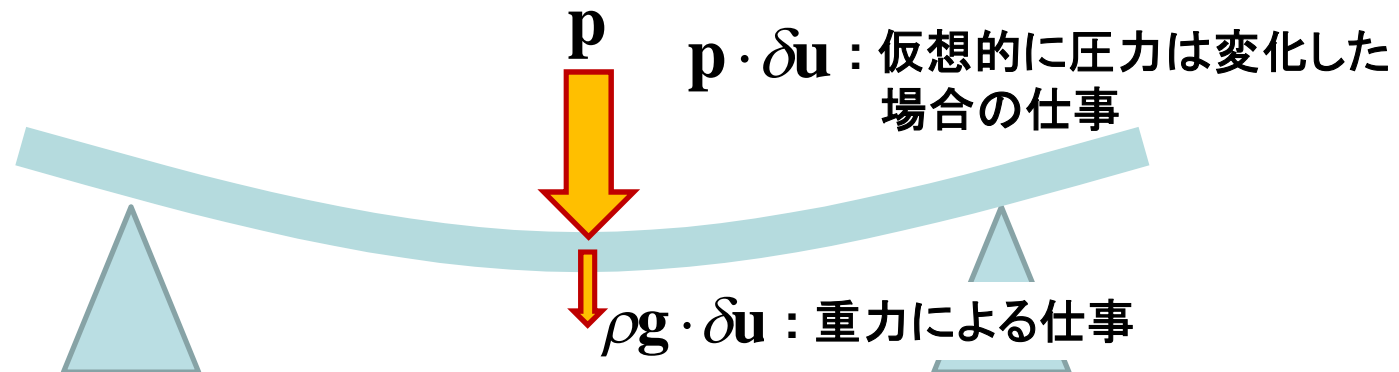
応力のエネルギーの仕事

表面から受ける外力

$$\int_V dV \sigma_{ij}^{(\varepsilon)} \delta \varepsilon_{ij} = \int_S dS (\mathbf{p} \cdot \delta \mathbf{u}) + \int_V dV (\bar{\mathbf{X}} \cdot \delta \mathbf{u})$$

物体全体

内部に直接受ける外力(重力、磁力)



🍵 応力の行列の固有値(主応力)を求めて対角化

$$\sigma_{ij}^{(\varepsilon)} = U^{-1} \begin{pmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & \sigma_2 \end{pmatrix} U$$

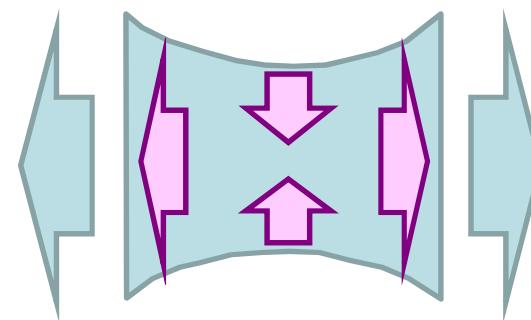
対称行列の固有値は、全て実数

🍵 U で対角化された座標系で見ると、
軸にそって物体は引っ張られる

🍵 + : 引く

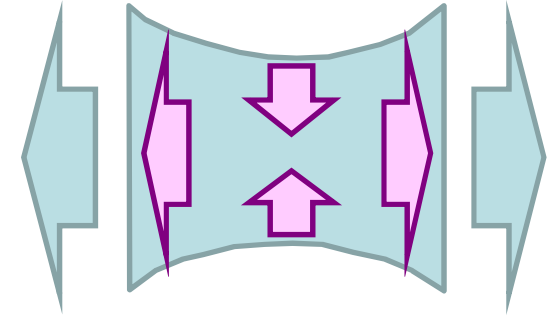
🍵 - : 押す

🍵 せん断はない



🍵 主応力が大きな時に断裂

$$\sigma_{ij}^{(\varepsilon)} = U^{-1} \begin{pmatrix} \sigma_0 & & \\ & \sigma_1 & \\ & & \sigma_2 \end{pmatrix} U$$

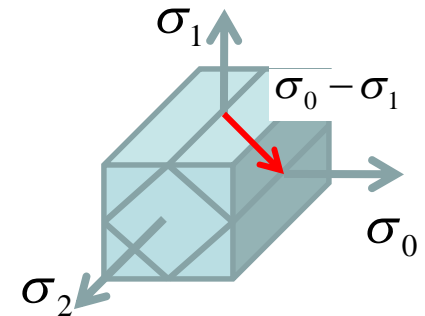


🍵 最大主応力説

$$\sigma_B = \max(\sigma_0, \sigma_1, \sigma_2)$$

🍵 最大せん断応力説 (トレスカの説)

$$\tau_B = \max \left(\left| \frac{\sigma_0 - \sigma_1}{2} \right|, \left| \frac{\sigma_1 - \sigma_2}{2} \right|, \left| \frac{\sigma_2 - \sigma_0}{2} \right| \right)$$



🍵 せん断ひずみエネルギー説 (ミーゼスの説)

$$2(\sigma_B)^2 = (\sigma_0 - \sigma_1)^2 + (\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_0)^2$$

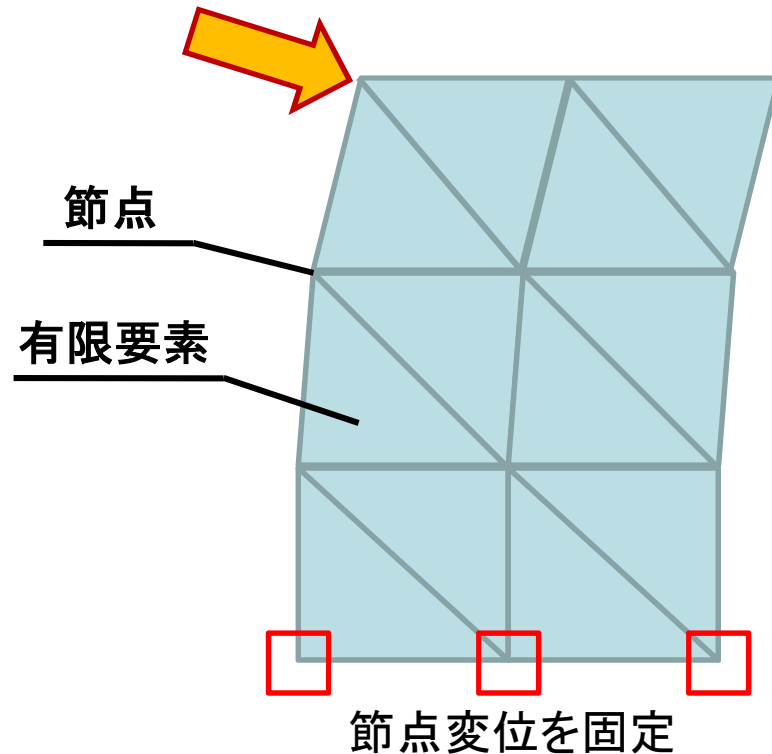
🍵 分離テンソル

🍵 主応力の分散を閾値と比較

🍵 物体を要素に分解して解析

🍵 4面体に分解

力を等価節点力として作用



🍵 マテリアル座標: $\mathbf{m}_{[0]}$, $\mathbf{m}_{[1]}$, $\mathbf{m}_{[2]}$, $\mathbf{m}_{[3]}$

🍵 ワールド座標: $\mathbf{p}_{[0]}$, $\mathbf{p}_{[1]}$, $\mathbf{p}_{[2]}$, $\mathbf{p}_{[3]}$

🍵 物体内の点

$$\mathbf{u} = \begin{bmatrix} \mathbf{m}_{[0]} & \mathbf{m}_{[1]} & \mathbf{m}_{[2]} & \mathbf{m}_{[3]} \end{bmatrix} \begin{bmatrix} b_{[0]} \\ b_{[1]} \\ b_{[2]} \\ b_{[3]} \end{bmatrix} = \mathbf{m}_{[i]} b_{[i]} \quad \mathbf{x} = \begin{bmatrix} \mathbf{p}_{[0]} & \mathbf{p}_{[1]} & \mathbf{p}_{[2]} & \mathbf{p}_{[3]} \end{bmatrix} \begin{bmatrix} b_{[0]} \\ b_{[1]} \\ b_{[2]} \\ b_{[3]} \end{bmatrix} = \mathbf{p}_{[i]} b_{[i]}$$

$b_{[i]} = 1$	➔	$\mathbf{u} = \mathbf{m}_{[i]}$
それ以外 = 0		$\mathbf{x} = \mathbf{p}_{[i]}$

🍵 ローカル座標: $\mathbf{u} = \mathbf{m}_{[i]} b_{[i]}$

🍵 ワールド座標: $\mathbf{x} = \mathbf{p}_{[i]} b_{[i]}$

$$b_{[i]} = \mathbf{m}_{[i]}^{-1} \mathbf{u}$$

$$\mathbf{x} = \mathbf{p}_{[i]} b_{[i]} = \mathbf{p}_{[i]} \mathbf{m}_{[i]}^{-1} \mathbf{u}$$

オブジェクト内の位置から、
モデルの形状と物体の位置を通して
ワールド座標での位置が分かる

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial x_i}{\partial u_j} + \frac{\partial x_j}{\partial u_i} \right)$$

$$\frac{\partial \mathbf{x}}{\partial u_i} = \mathbf{p}_{[j]} \mathbf{m}_{[j]}^{-1} \frac{\partial \mathbf{u}}{\partial u_i} = \mathbf{p}_{[j]} \mathbf{m}_{[j]}^{-1} \boldsymbol{\delta}_i \quad \boldsymbol{\delta}_i = [\delta_{i0} \quad \delta_{i1} \quad \delta_{i2}]^T$$

$$\varepsilon_{ij} = \frac{1}{2} \left(\mathbf{p}_{[k]} \mathbf{m}_{[k]}^{-1} + \left(\mathbf{p}_{[k]} \mathbf{m}_{[k]}^{-1} \right)^T \right)_{ij} = \mathbf{B}_{ik} \mathbf{p}_{[k]j}$$

要素内の場所に依存しない = 要素内ですべて同じ

応力も要素内ですべて同じ

$$\int_V dV \boldsymbol{\sigma}^{(\varepsilon)} : \delta \boldsymbol{\varepsilon} = \int_S dS (\mathbf{p} \cdot \delta \mathbf{u}) + \int_V dV (\bar{\mathbf{X}} \cdot \delta \mathbf{u})$$

四面体分割

物体内部の面は
接する面の力が互いに打ち消す

$$\sum_{\text{全ての四面体}} \left\{ \int_{\text{四面体内}} dV \boldsymbol{\sigma}^{(\varepsilon)} : \delta \boldsymbol{\varepsilon} - \int_{\text{表面三角形}} dS (\mathbf{p} \cdot \delta \mathbf{u}) - \int_{\text{四面体内}} dV (\bar{\mathbf{X}} \cdot \delta \mathbf{u}) \right\} = 0$$

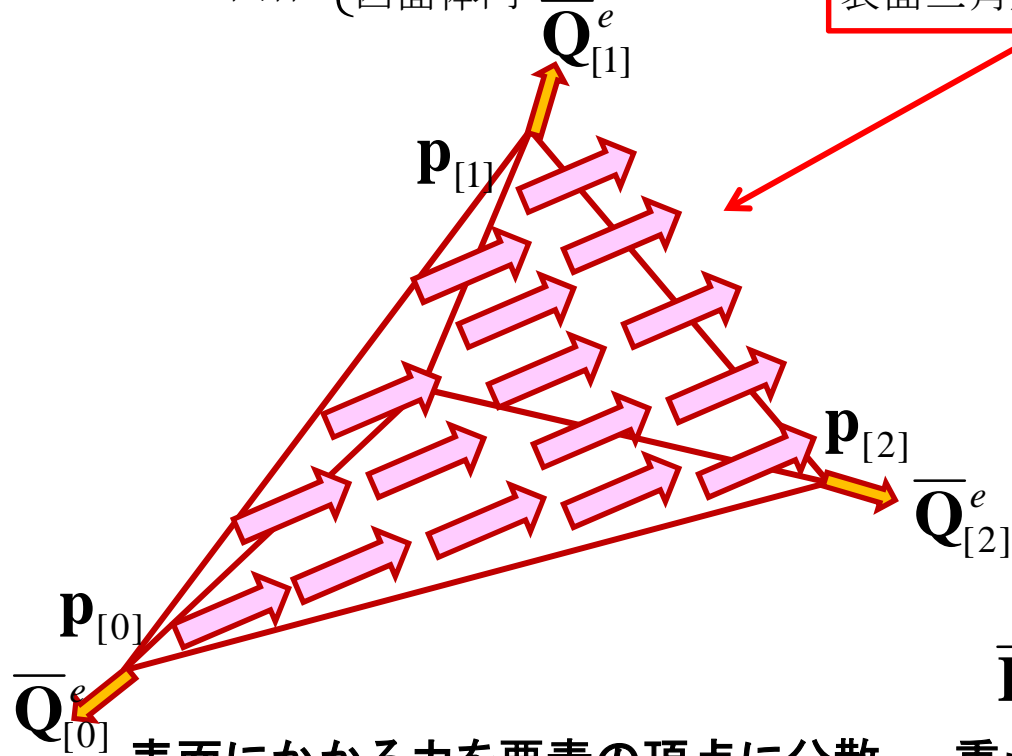
$$\left(\int_{\text{四面体内}} dV \sigma_{ij}^{(\varepsilon)} B_{ik} \right) \delta \mathbf{p}_{[k]j} = k_{ijkl}^e \mathbf{p}_{[l]i} \delta \mathbf{p}_{[k]j}$$

要素剛性マトリックス

外力の四面体分割化



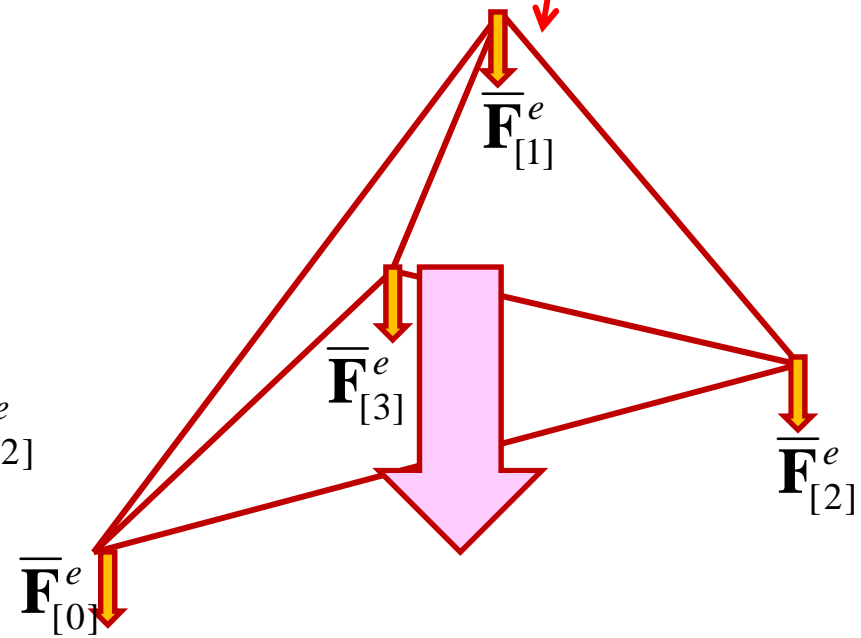
$$\sum_{\text{全ての四面体}} \left\{ \int_{\text{四面体内}} dV \sigma_{ij}^{(\varepsilon)} \delta \varepsilon_{ij} - \int_{\text{表面三角形}} dS (\mathbf{p} \cdot \delta \mathbf{u}) - \int_{\text{四面体内}} dV (\bar{\mathbf{X}} \cdot \delta \mathbf{u}) \right\} = 0$$



表面にかかる力を要素の頂点に分散

$$\int_{\text{表面三角形}} dS (\mathbf{p} \cdot \delta \mathbf{u}) = \bar{\mathbf{Q}}_{[k]j}^e \delta \mathbf{p}_{[k]j}$$

表面三角形



重心にかかる力を計算し要素の頂点に分散

$$\int_{\text{四面体内}} dV (\bar{\mathbf{X}} \cdot \delta \mathbf{u}) = \bar{\mathbf{F}}_{[k]j}^e \delta \mathbf{p}_{[k]j}$$

四面体内

$$\sum_{\text{全ての四面体}} \left\{ k_{ijkl}^e \mathbf{p}_{[l]i} \delta \mathbf{p}_{[k]j} - \bar{\mathbf{Q}}_{[k]j}^e \delta \mathbf{p}_{[k]j} - \bar{\mathbf{F}}_{[k]j}^e \delta \mathbf{p}_{[k]j} \right\} = 0$$

- ☕ 頂点の位置は自由に動かせることができる
= 和の中身は全て0でないといけない

$$k_{ijkl}^e \mathbf{p}_{[l]i} - \bar{\mathbf{Q}}_{[k]j}^e - \bar{\mathbf{F}}_{[k]j}^e = 0 \quad : \text{要素の剛性方程式}$$

- ☕ 隣接している要素で共有された頂点にかかる力は各要素における力の和

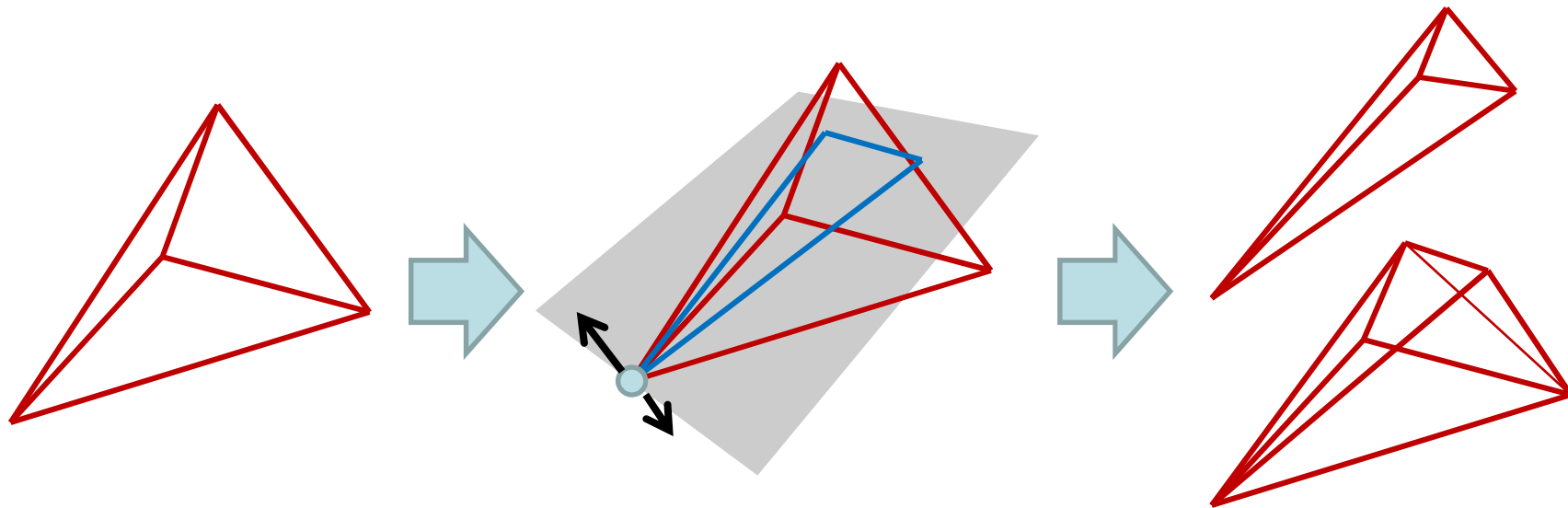
$$\sum_{\text{四面体}} k_{ijkl}^e \mathbf{p}_{[l]i} - \sum_{\text{四面体}} \bar{\mathbf{Q}}_{[k]j}^e - \sum_{\text{四面体}} \bar{\mathbf{F}}_{[k]j}^e = 0 \quad : \text{全体の剛性方程式}$$

- ☕ 要素の剛性方程式だけで解ける部分(固定端、力が掛からない)を先に解き、その後、全体の剛性方程式でないといけない部分を解く

- ☕ 頂点を共有している要素の数はドロネー分割では $O(1)$

- 🍵 頂点の力の和に関する主応力から分離することが決まれば、主応力等を法線とする平面4面体で分割

$$\sigma_{ij}^{(\varepsilon)} = \lambda (\text{Tr } \varepsilon_{ij}) \delta_{ij} + 2\mu \varepsilon_{ij}$$



🍵 細かなモデルを作成
しておき、重心が入る
要素の位置、向きで描画

🍵 メリット

🍵 メモリ使用量に
上限がつけやすい

🍵 欠点

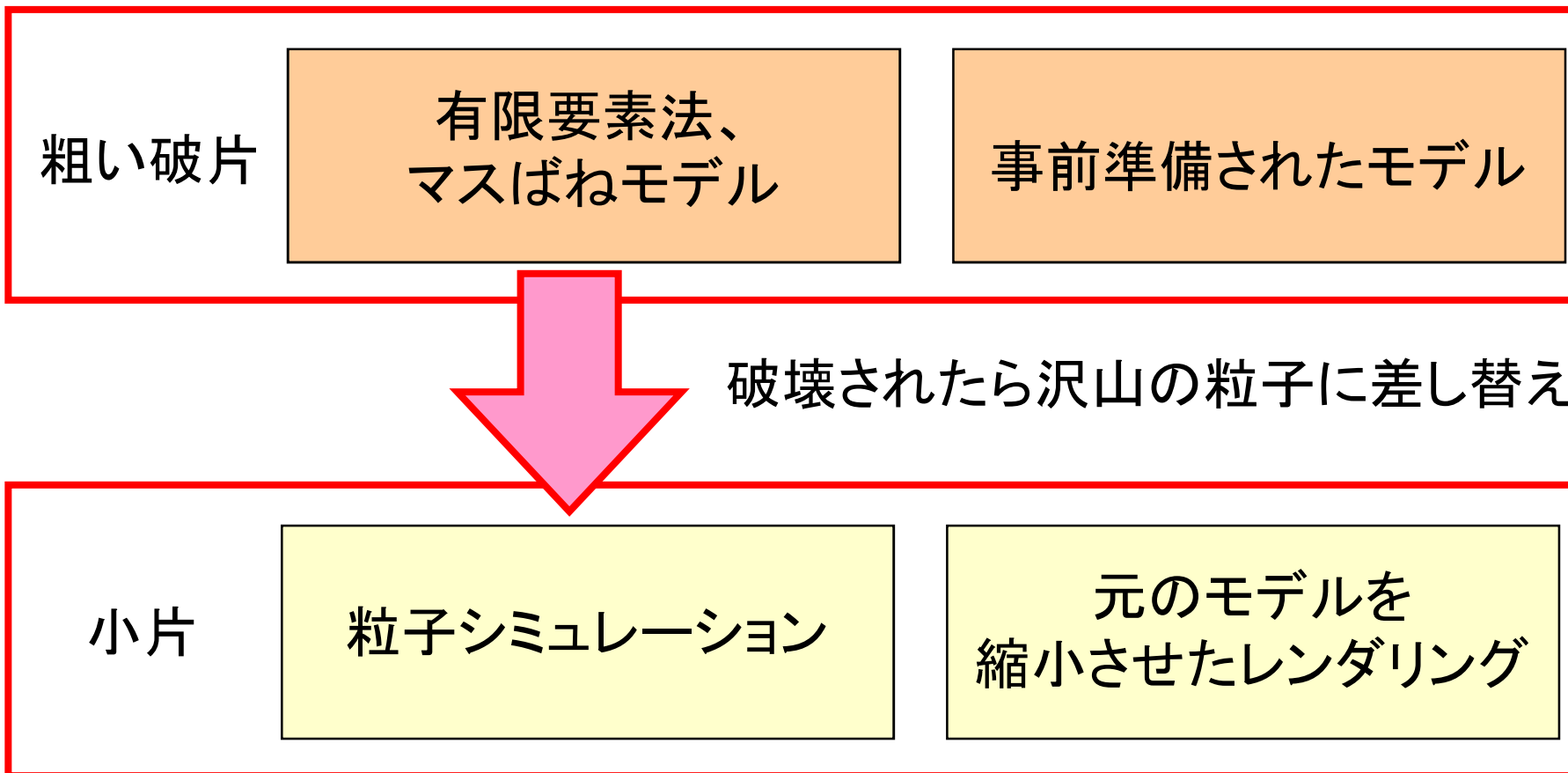
🍵 作成したモデル以上の
細かさが実現できない

- 🍵 はじめに
- 🍵 歴史
- 🍵 破壊シミュレーション
- 🍵 **細かな破壊の表現**
- 🍵 まとめ

- 🍵 非常に細かな破砕物をまじめに計算するのは大変
- 🍵 ある程度大きな力を受けたら小さなパーティクルで表現すれば良いんじゃない？
- 🍵 それぞれはどんな大きさにすればよい？
 - 🍵 全てが同じ大きさなのはダサい
 - 🍵 現実には、破砕物はどんな大きさの分布をしているの？

物理シミュレーション

レンダリング



🍵 粉体工学で幾つかの分布関数が知られている

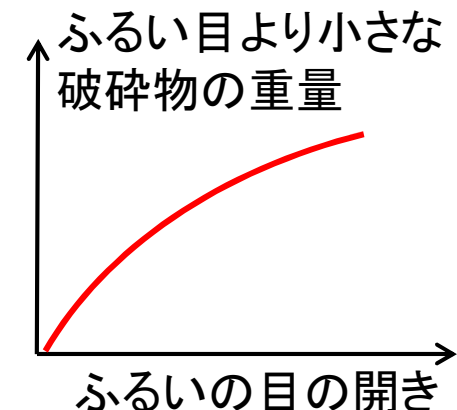
🍵 ふるいの目より小さな質量（ふるいの下に落ちた物体の重さ）の関数

🍵 Gaudin-Schuhmann分布 [Gaudin 1926, Schuhmann 1980]

🍵 粗砕(数10cm-数10mm)ないし中砕(数10cm-数10 μ m)程度の粉碎物に当てはまることが多い

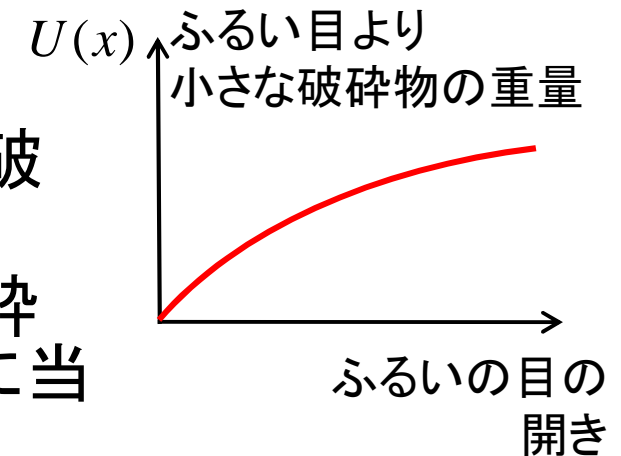
🍵 Rosin-Rammler 分布 [Rosin and Rammler 1933]

🍵 石炭粉碎物のふるいの目より大きな質量の量を表すのに用いる粒度分布



🍵 Gaudin-Schuhmann分布

- 🍵 ふるいの目より小さな質量を持つ破砕物に関する粒度分布
- 🍵 粗砕(数10cm-数10mm)ないし中砕(数10cm-数10 μ m)程度の粉碎物に当てはまることが多い



$$U(x) = \left(\frac{x}{R_e} \right)^m$$

R_e : 粒度特性数
 m : 均等数

粒度分布は一般に次のように書くことができる

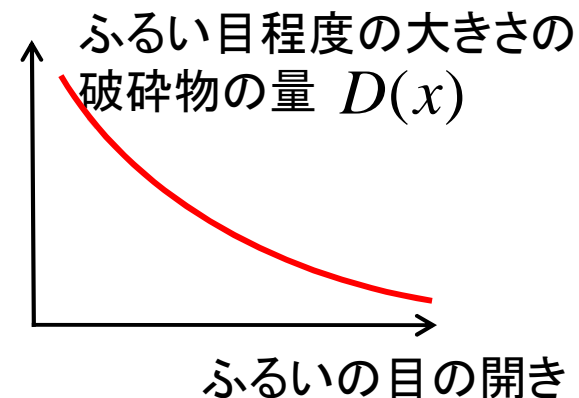
$$U(x) = \int \rho x^3 D(x) dx \quad D(x) : \text{半径}x\text{の破砕物の数}$$

2つの式を微分すると

$$U'(x) = \rho x^3 D(x) \equiv m \frac{x^{m-1}}{R_e^m} \quad \Rightarrow \quad D(x) = \frac{m}{\rho R_e^m} x^{-(4-m)}$$

🍵 粒子数分布と パーティクルシミュレーション

$$U(x) = \int \rho x^3 D(x) dx \quad D(x) = Ax^{-(4-m)}$$
$$A = \frac{m}{\rho R_e^m}$$



R_{dust} から R_{max} の大きさの破砕物をN個の粒子で表現する

$$U(x) = \int \rho x^3 D(x) dx \quad \longrightarrow \quad U(x) \approx \sum_{i=1}^N \rho r_i^3 + U(R_{dust})$$

同じ近似として、次の式が成り立つ。

$$N(x) = \sum_{i=1}^N 1 \approx \int_{R_{dust}} D(x) dx$$

ゆえに、次の式を解いて、i番目の粒子の大きさを求める

$$\frac{i}{N} N(R_{max}) \equiv N(r_i) = \int_{R_{dust}}^{r_i} D(x) dx$$

🍵 以上の式を解くと...

$$N(x) = \int_{R_{dust}} D(x) dx = \int_{R_{dust}} Ax^{-(4-m)} dx = -\frac{A}{3-m} x^{-(3-m)} \Big|_{R_{dust}}$$

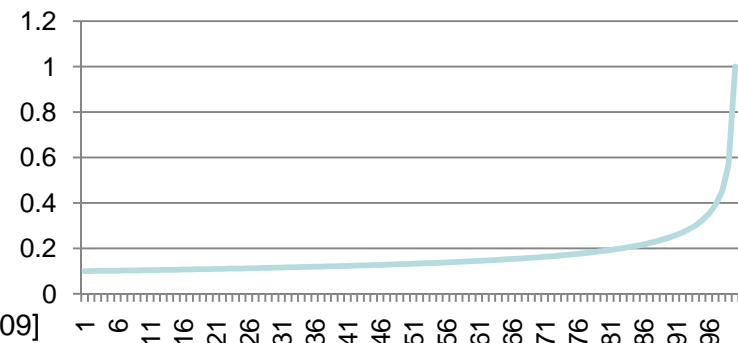
$$= \frac{A}{3-m} \left[R_{dust}^{-(3-m)} - x^{-(3-m)} \right]$$

$$\frac{i}{N} N(R_{max}) \equiv N(r_i)$$

$$\Rightarrow \frac{i}{N} \frac{A}{3-m} \left[R_{dust}^{-(3-m)} - R_{max}^{-(3-m)} \right] = \frac{A}{3-m} \left[R_{dust}^{-(3-m)} - r_i^{-(3-m)} \right]$$

$$r_i = \frac{R_{max} R_{dust}}{\left[R_{max}^{3-m} - \frac{i}{N} \left(R_{max}^{3-m} - R_{dust}^{3-m} \right) \right]^{\frac{1}{3-m}}}$$

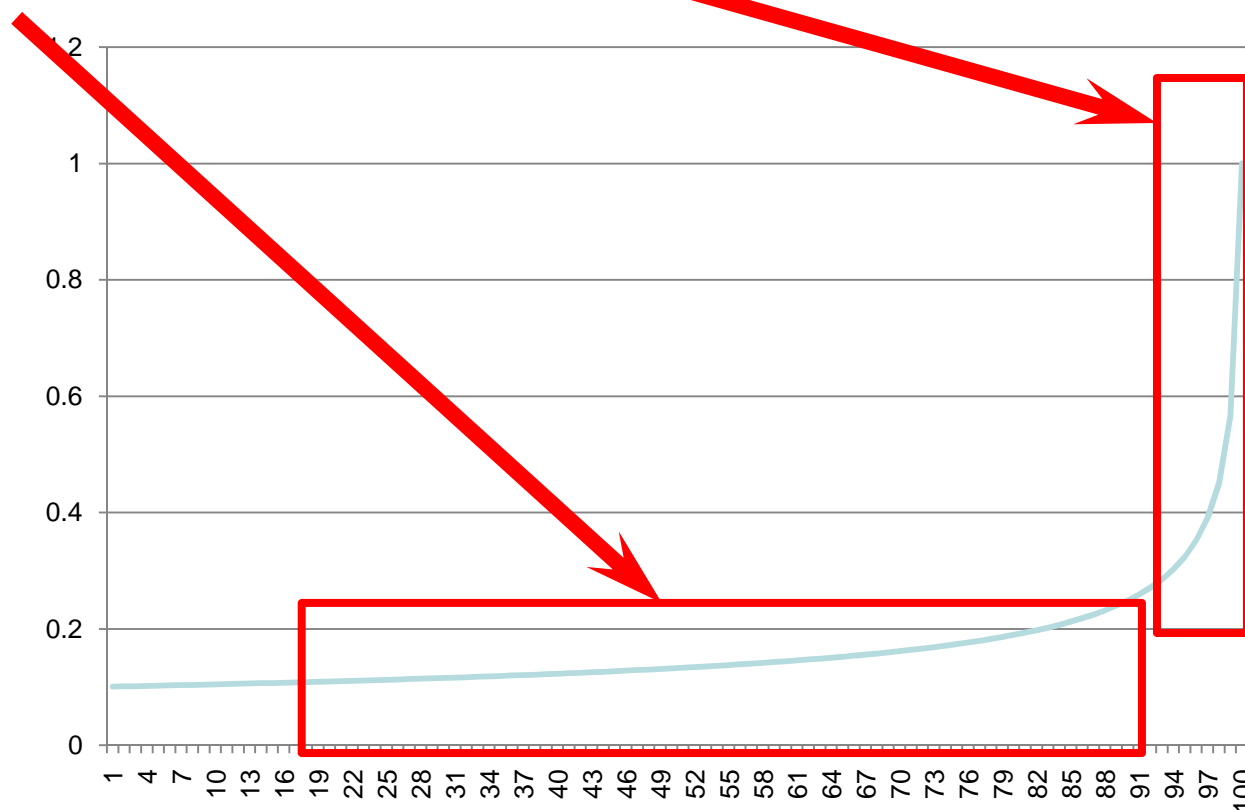
[Imagine et al. 2009]



🍵 細かな破砕物をパーティクルとして表現

🍵 数個は大きめ

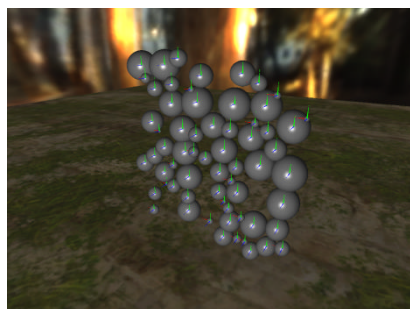
🍵 ほとんどの粒子は小さく



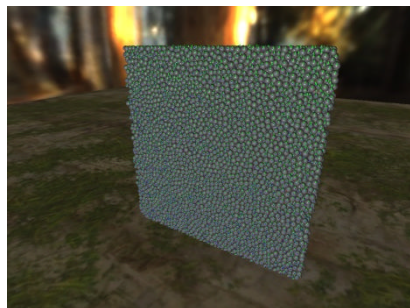
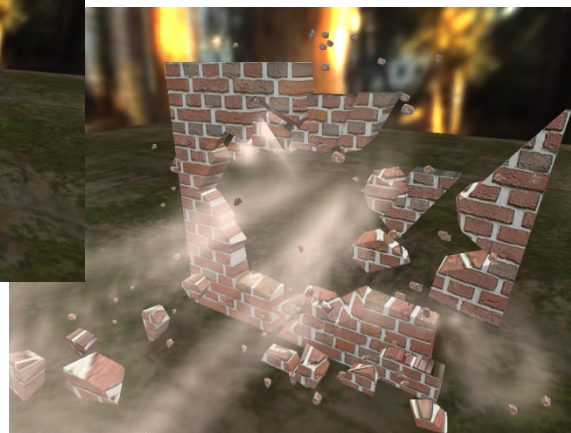
要素数の変更に伴うシミュレーション結果の違い



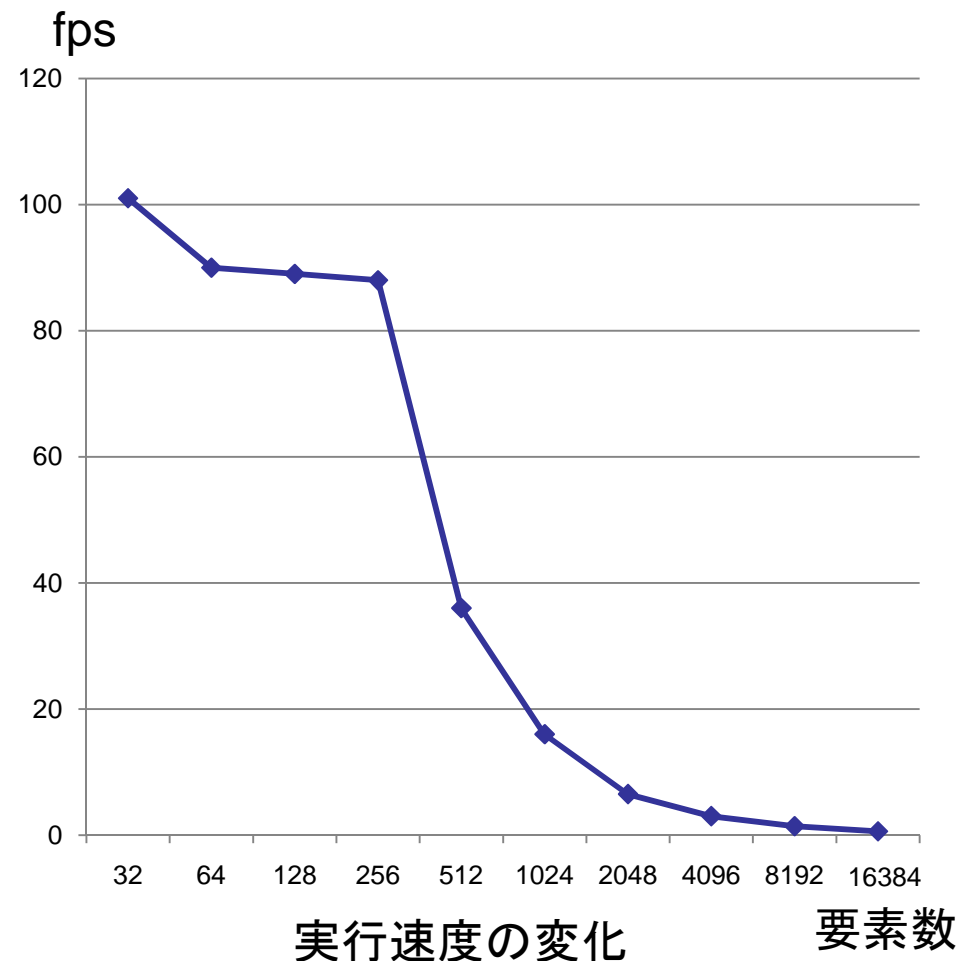
🍵 密度、限界応力を固定 $\rho = \frac{Volume}{Number} = \frac{4\pi}{3} Radius^3$, $\sigma = k\Delta x_{cutoff} / Radius^2$



64要素



16384要素



🍵 重力は大きい

- 🍵 正確な計算するなら構造力学を考慮してモデリングをする必要がある
- 🍵 固定している時には、重力が掛からないようにするのは(ゲーム的には)悪くないと思います

- 🍵 はじめに
- 🍵 歴史
- 🍵 破壊シミュレーション
- 🍵 細かな破壊の表現
- 🍵 **まとめ**

- 🍵 有限要素法を勉強してみました
- 🍵 細かな破片の表現をどうすれば良いか提案しました

🍵 剛体シミュレーションに似ている

🍵 なんとか作れそう

🍵 でも、市販されている物にたどりつくのは大変

🍵 そもそも面白い？

🍵 なんでも壊せるのはレベルデザインを難しくするだけでは...

🍵 かといって、壊せるものと壊せないものがあるのはダサいしね。

🍵 本当に破壊を生かしたゲームか
メリットが大きい場合以外は
作らなくていいんじゃない？

ご清聴ありがとうございました



🍵 質問はございませんか？