

接触と摩擦



Code_Aster, Salome-Meca course material

GNU FDL licence (<http://www.gnu.org/copyleft/fdl.html>)

概要

▶ 接触の定義

▶ 離散の式による、接触問題の解法

- ペアリング
- 方程式
- 計算説明

▶ 摩擦の定義

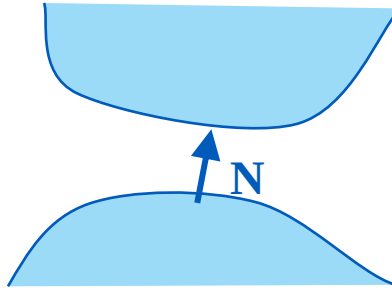
▶ 摩擦のある接触問題を解くための連続体ベースの式

▶ *Code_Aster* 使用法

▶ まとめ

一方向接触一定義 (1)

▶ 一方向の接触



▶ 接触での未知数:

- 2つの物体の間の符号付き距離(ギャップ)
- 接触圧

$$g = \mathbf{u} \cdot \mathbf{N}$$

$$p = \boldsymbol{\sigma} \cdot \mathbf{N} \cdot \mathbf{N}$$

一方向接触一定義(2)

▶ Signorini-Hertz-Moreau の条件

■ 幾何学的条件: 物体の« 食い込みなし » (Signorini)

$g \geq 0$ $g > 0$ 接触していない
 $g = 0$ 接触 (物体が連続している)

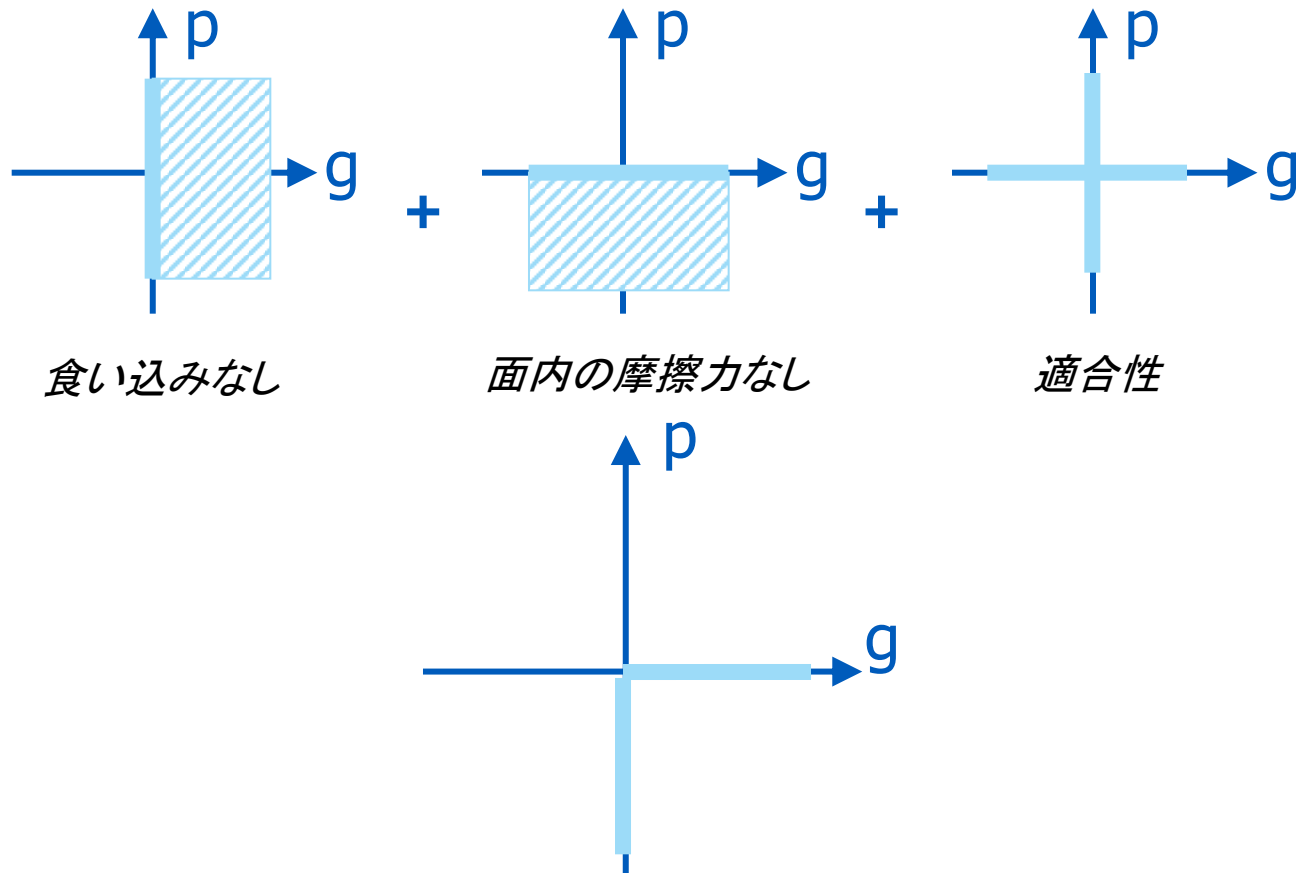
■ 機械的条件: « 接触圧 » (Hertz)

$p \leq 0$ $p = 0$ 接触していない
 $p < 0$ 接触 (圧縮されている)

■ エネルギー条件: 適合性(Moreau)

$p \cdot g = 0$ $p = 0$ 分離
 $g = 0$ 接触

一方向接触 – 定義 (3)



一方向接触一定義(4)

▶ 接触の取り扱い

- 一義的でない (関数で表現できない)
- $g=0$ では微分不可能

▶ 摩擦のない接触は可逆的かつ保存的

- 接触のポテンシャルエネルギーが存在
- 接触は、食い込む速度や食い込む経路に非依存
- 剛性塑性モデルとのアナログが可能

離散法

▶ 接触する可能性のある面の上の節点が互いに食い込まないという関係に基づく解法

- 接触面に関する離散的な表現 (有限要素法)
- 接触する可能性のある面の上の節点の投影
- 「アクティブな」節点の間の運動学的関係の定義
- 制約付き最適化に基づく解決アルゴリズム

▶ 2ペアリング法

- 節点の接近: 互換性のあるサーフェスメッシュに限定
- 節点と微小面の接近: 任意のサーフェスメッシュ

概要

▶ 接触の定義

▶ 離散の式による、接触問題の解法

■ ペアリング

■ 方程式

■ 計算方法

▶ 摩擦の定義

▶ 摩擦のある接触問題を解くための連続体ベースの式

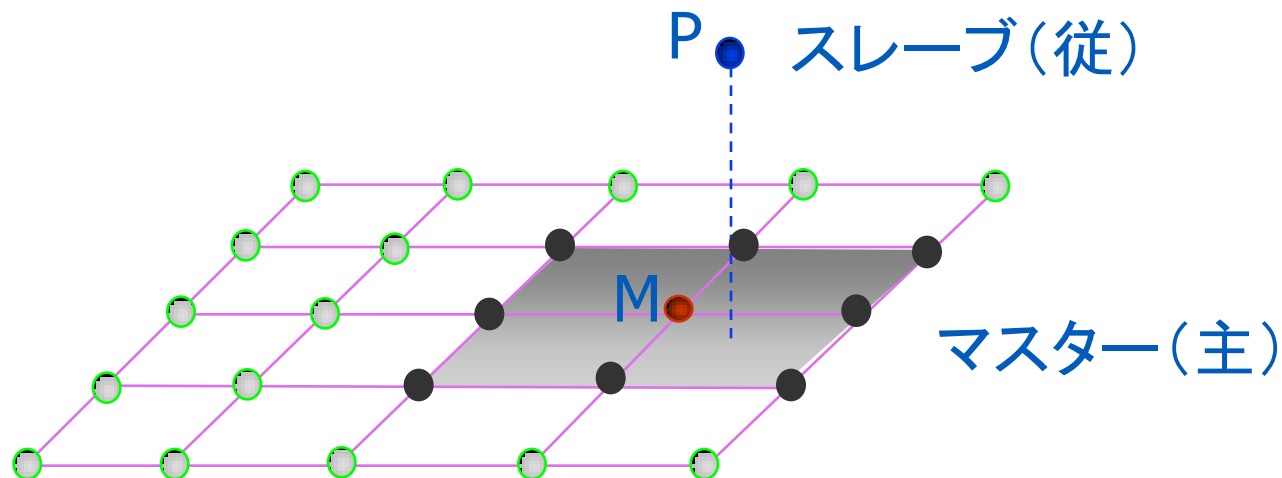
▶ *Code_Aster* の使用法

▶ まとめ

幾何学的なペアリング(1)

◆ 節点と微小面のペアリング

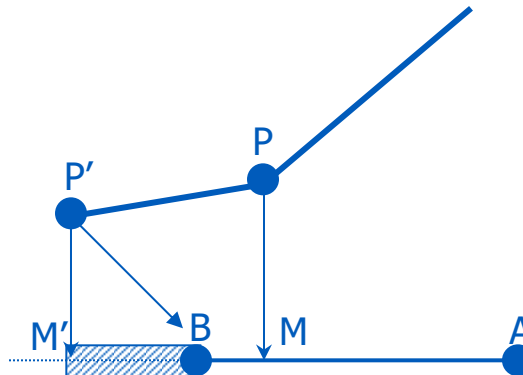
- スレーブ節点P ● に最も近いマスター節点M ● の探索 (距離の計算)
- マスター面上の節点のループ



幾何学的なペアリング(2)

▶ 節点と微小面のペアリング (前頁より)

- 最も近いマスターの節点に連結されているマスターの要素を照合する
 - 節点 **M** への投影と距離を計算する
 - 現在ペアになっているマスタの微小面 (節点 **M'** と境界 **B**) の外に投影された時は、投影を修正する
キーワード: `TOLE_PROJ_EXT`
 - 微小面の選択: **PM** の距離が最小になるように
- 隙間の定義: **PM.N**
 - 最初の隙間を計算する (**PM.N**)
 - どの法線を用いるか? マスター、スレーブ、それとも 平均
 - **M** のパラメトリック座標系



幾何学的なペアリング (3)

▶ ペアリングの更新

- ニュートン法での収束計算が終了したらすぐに更新
- 接触の幾何学的非線形性に依存
- 更新の回数
 - プログラムで選択：自動で回数を固定
 - ユーザーによる選択
 - なし (最初の設定を保つ、微小な摂動の場合のみ)

▶ 注意! 幾何学的な更新の失敗が (相互の食い込みによる) 誤った結果を生じ得る

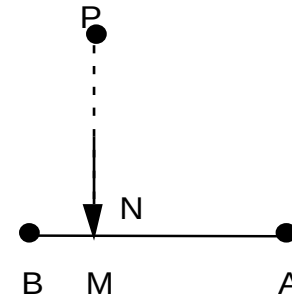
▶ プログラムによる設定...

例 :運動的關係(1)

▶ 食い込みを起こさない運動学的関係

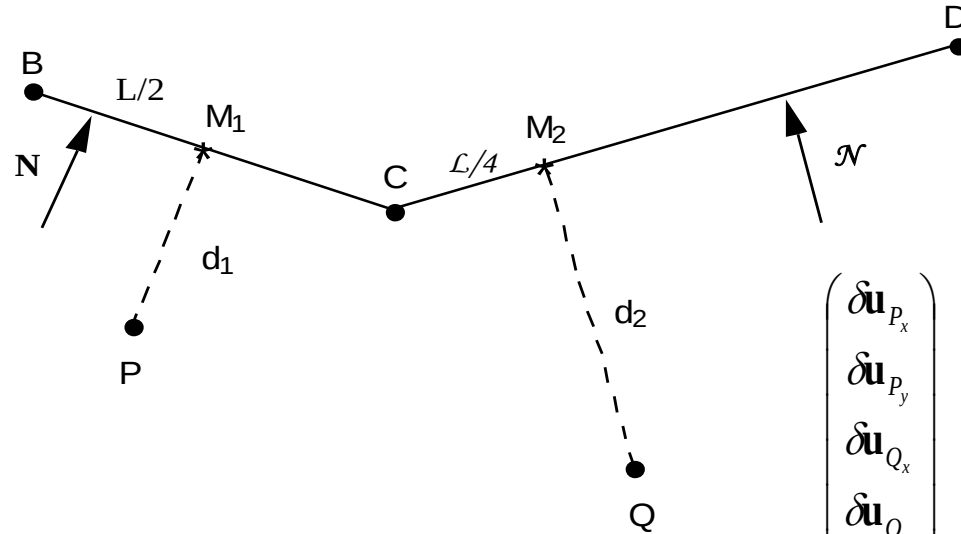
- スレーブの節点 P とそのPからマスター面への投影M との関係
- 節点M の変位は、形状関数とパラメトリック座標系を経由した要素の変位に関連づける

$$\left[\delta \mathbf{u}_P - \sum_{j=1}^{n_{mat}} \Phi_{B_j}(M) \delta \mathbf{u}_{B_j} \right] \cdot \mathbf{N} \leq d^-$$



例 :運動的關係(2)

▶ 例



$$\begin{pmatrix} N_x & N_y & 0 & 0 & -0.5N_x & -0.5N_y & -0.5N_x & -0.5N_y & 0 & 0 \\ 0 & 0 & \mathcal{N}_x & \mathcal{N}_y & 0 & 0 & -0.75\mathcal{N}_x & -0.75\mathcal{N}_y & -0.25\mathcal{N}_x & -0.25\mathcal{N}_y \end{pmatrix} \begin{pmatrix} \delta u_{P_x} \\ \delta u_{P_y} \\ \delta u_{Q_x} \\ \delta u_{Q_y} \\ \delta u_{B_x} \\ \delta u_{B_y} \\ \delta u_{C_x} \\ \delta u_{C_y} \\ \delta u_{D_x} \\ \delta u_{D_y} \end{pmatrix} \leq \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

$\swarrow$

A

$\swarrow$

u

$\downarrow$

d

概要

- ▶ 接触の定義
- ▶ 離散の式による、接触問題の解法
 - ペアリング
 - 方程式
 - 計算方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

方程式 (1)

▶ ラグランジュ乗数の使用(二重化, 線形システムサイズの増加)

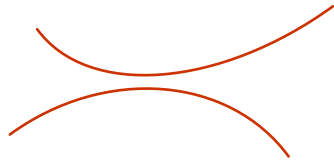
- つり合い方程式
- 食い込みなし
- 横向き之力なし
- 適合性

$$\left\{ \begin{array}{l} \mathbf{K} \cdot \mathbf{u} + \mathbf{A}^T \cdot \boldsymbol{\lambda} = \mathbf{F} \\ \mathbf{A} \cdot \mathbf{u} \leq d^- \\ \boldsymbol{\lambda} \geq 0 \\ (\mathbf{A} \cdot \mathbf{u} - d^-)_j \cdot \lambda_j = 0, \forall j \end{array} \right.$$

方程式 (2)

- ▶ ペナルティ関数の使用(線形システムの規模は不変)

$$\mathbf{K} \cdot \mathbf{u} + \frac{1}{\eta} \mathbf{A}^T \cdot [\mathbf{A} \cdot \mathbf{u} - \mathbf{d}]^+ = \mathbf{F}$$



0 = 剛性



$1/\eta$ = 法線方向の剛性

$$\text{アナロジ : } \mu = \frac{1}{\eta} [\mathbf{A} \mathbf{u} - \mathbf{d}]^+$$

概要

- ▶ 接触の定義
- ▶ 離散の式による、接触問題の解法
 - ペアリング
 - 方程式
 - 計算方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

アクティブ・セット・アルゴリズム (1)

▶ `ALGO_CONT = 'CONTRAINTE'`

- 摩擦の無い接触問題でのデフォルトのアルゴリズム
- 拘束された凸面の最適化に由来する ; 接触がない問題の解が必要
- 有限回数の反復計算による収束

▶ アルゴリズム

- 食い込みがあれば制約を有効にしてギャップをゼロに設定
- 有効となった制約を含むサブシステムを計算

$$\begin{cases} \mathbf{K} \cdot \mathbf{u} + \mathbf{A}^T \cdot \boldsymbol{\lambda} = \mathbf{F} \\ \mathbf{A} \cdot \mathbf{u} = d^-_{actives} \end{cases}$$

アクティブ・セット・アルゴリズム (2)

◆ アルゴリズム (前頁からの続き)

- 変位 u と ラグランジュ乗数 λ を計算する
- 食い込みの有無を確認
 - $A \cdot u > d$ ならば, 制約がアクティブでないため食い込みが発生
 - 最大の食い込みを発生させている部分の制約をアクティブにした後, 再び計算を反復
- 接触面内の摩擦力が有無を確認
 - $\lambda > 0$ ならば, 摩擦力が発生
 - 最小の摩擦力を生じる制約を非アクティブにした後, 再び計算を反復
- 制約の追加と削除がなければ, 問題は収束する. そうでなければ反復

アクティブ・セット・アルゴリズム (3)

▶ アクティブ・セット法の反復解法版 (`ALGO_CONT= 'GCP'`)

- 多数の接触節点がある問題に対しては非常に効率的
- 摩擦の無い問題のみ

▶ すべり拘束 (`GLISSIERE = 'OUI'`)

- 双方向接触：2つの面が互いに滑って離れない
- 横向きの力が存在する
- 摩擦の無い問題のみ

代替アクティブセットアルゴリズム

▶ `ALGO_CONT = 'LAGRANGIEN'`

- アクティブセットアルゴリズムと同じ考え
ただし、複数の制約が(1個ずつでなくて)一度に追加または削除される
- アクティブセットアルゴリズムより速い場合がある

ペナルティ法

▶ `ALGO_CONT = 'PENALISATION'`

- 接触を考慮せずに問題を解決

- 左辺と右辺に項を追加

 - 接線方向の剛性が必要

$$\mathbf{K}\mathbf{u} + E_N \mathbf{A}^T [\mathbf{A}\mathbf{u} - \mathbf{d}^-]^+ = \mathbf{F}$$

- ペナルティ関数の選択は非常に重要
`E_N`に対する感度解析が必要

適切なアルゴリズムの選択

▶ 双対的な手法 (ラグランジュ乗数法)

- “正確” な方法
- 食い込みなし
- ユーザーが設定すべき係数が無い
- 接触力の精度が良い
- レガシーアクティブステート法は, 接触する節点の数が少ない場合が好ましい
- 反復アクティブステート法は, 接触する節点の数が多い場合が好ましい

▶ 初期の規則的な手法 (ペナルティ法)

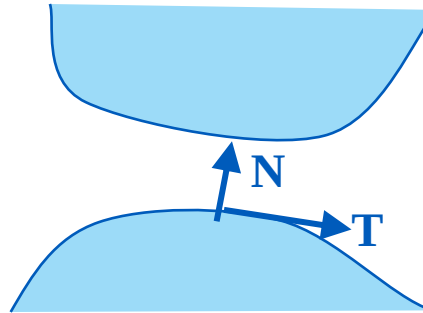
- 近似の方法
- 解がペナルティ係数に対して強く依存
- Dirichlet 条件を妨げない

概要

- ▶ 接触の定義
- ▶ 離散の式による、接触問題の解法
 - ペアリング
 - 方程式
 - 解決方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

接触/摩擦 – 定義 (1)

- ◆ 2つの物体の間での摩擦がある一方向接触



- ◆ 2D(2次元)では、1つの接線ベクトルが必要
- ◆ 3D(3次元)では、接平面を定義するために2つのベクトル(t_1 と t_2)が必要

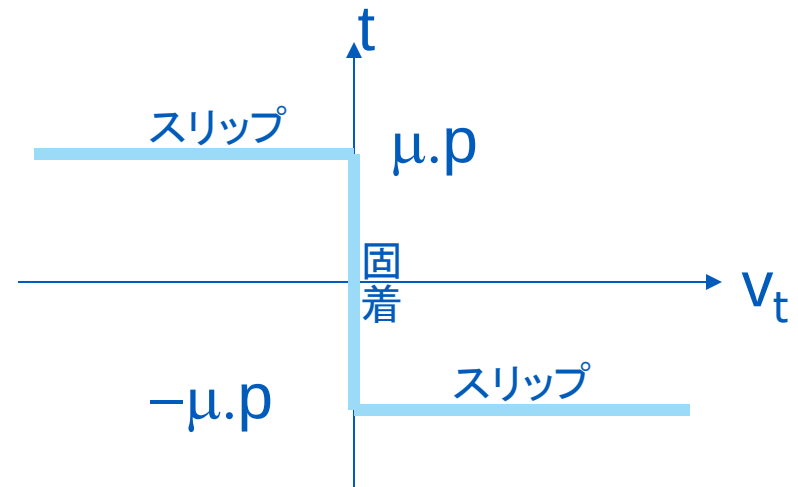
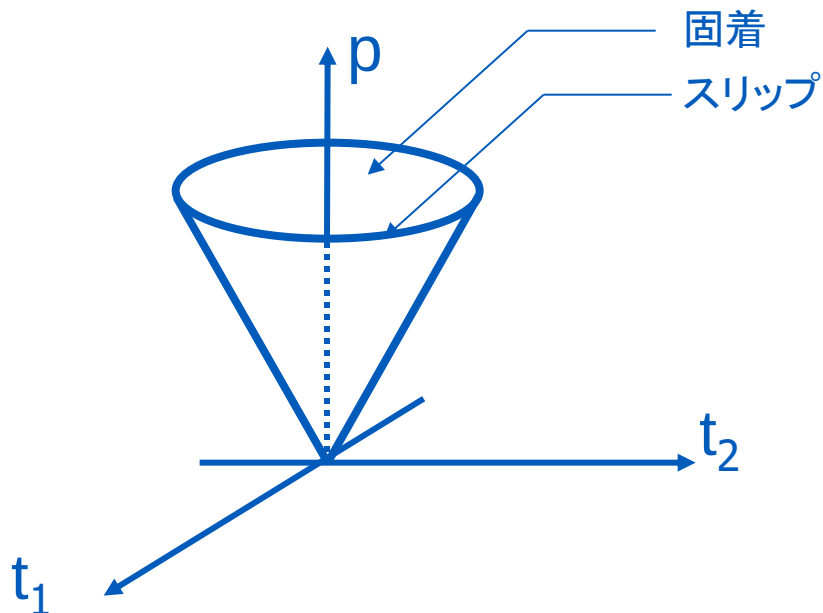
接触/摩擦 – 定義 (2)

- ◆ 摩擦接触: スリップ発生なし(固着状態)が可能な力のしきい値 f
- ◆ 第1の固体層が第2の固体層に及ぼすせん断力を t とする
- ◆ 塑性変形が始まるしきい値のアナロジから, スリップが始まるしきい値は接線 t の値に依存して, 以下のように書ける:
$$\phi(t) \leq 0$$

■ クーロン摩擦: $\phi(t) = ||t|| - \mu |p|$ ここに p は接触圧力、 μ は静摩擦係数

接触/摩擦 – 定義 (3)

摩擦円錐



接触/摩擦 – 定義 (4)

◆ 接触の定式化

$$\begin{cases} u_n \leq g \\ p \leq 0 \\ p \cdot (u_n - g) = 0 \end{cases}$$

◆ 摩擦の定式化

$$\begin{cases} \|\mathbf{t}\| - \mu |p| \leq 0 \\ \mathbf{v}_t = \lambda_g \cdot \frac{\mathbf{t}}{\|\mathbf{t}\|} \\ \lambda_g \cdot (\|\mathbf{t}\| - \mu |p|) = 0 \\ \lambda_g \leq 0 \end{cases}$$

接触/摩擦 – 定義 (5)

▶ クーロン摩擦での力/速度の関係

- $p < 0$ ならば $g = 0$, (接触状態) ここに, $\phi(\mathbf{t}) = \|\mathbf{t}\| - \mu \cdot |p|$
- $\phi(\mathbf{t}) < 0$ ならば, $\mathbf{v}_t = 0$ (固着状態)
- $\phi(\mathbf{t}) = 0$ ならば, $\mathbf{v}_t = \lambda_\gamma \cdot \mathbf{t}$ かつ $\lambda_\gamma \leq 0$ (すべり状態)

接触/摩擦 – 定義 (6)

- ▶ クーロン摩擦を伴う一方向接触は不可逆的かつ非保存的である
 - 摩擦のポテンシャルエネルギーは定義できない
- ▶ 摩擦は、接線方向の速度に関連して、変位には関連しない（摩擦のない接触との違い!）

ペナルティ法

▶ `ALGO_FROT = 'PENALISATION'`

- 摩擦接触でのデフォルトの方法
- 接触状態を, ペラグランジュ乗数 (`ALGO_CONT = 'LAGRANGIEN'`) とペナルティを伴う摩擦によって扱う.
- 摩擦によるペナルティの係数を `E_T` で設定

規則のアルゴリズム

▶ `ALGO_FROT = 'LAGRANGIEN'`

- ラグランジュ乗数はアクティブセット戦略によって接触と摩擦の両方に使用される
- 2D(2次元) と 3D(3次元) では, 収束についての理論的な知見は無い.
- 2D(2次元)では、 $\lambda_g = \pm \alpha \cdot t$ という式から, アルゴリズムは非常に簡単で効率的になる.

適切なアルゴリズムの選択

◆ ラグランジュ乗数法

- 正確
- 摩擦接触の自由度が多い (特に3Dの場合)
- 収束が遅い (特に3Dの場合)
- ディリクレ境界条件に非常に敏感
- 接線方向の剛性に余分な項が加わっている(3D の場合のみ)

◆ ペナルティ法

- 近似
- 正規化項の値に非常に敏感
- 収束が遅い
- Dirichlet境界条件にあまり影響されない
- 接線方向の剛性に余分な項が加わっている(2Dと3Dの場合)

◆ 摩擦接触には、常に**CONTINUE**メソッドを使用すること (次頁で詳しく説明)

- ▶ 接触摩擦についての離散の定式化は ,
Code_Aster の(15年前に登場した)古い計算手法
- ▶ 摩擦接触の問題は数学的に困難
高度なツールを含む研究領域
- ▶ 現在の手法は, «連続»の式に基づく

概要

- ▶ 摩擦の定義
- ▶ 離散的定式化による接触問題の解法
 - ペアリング
 - 方程式
 - 解決方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

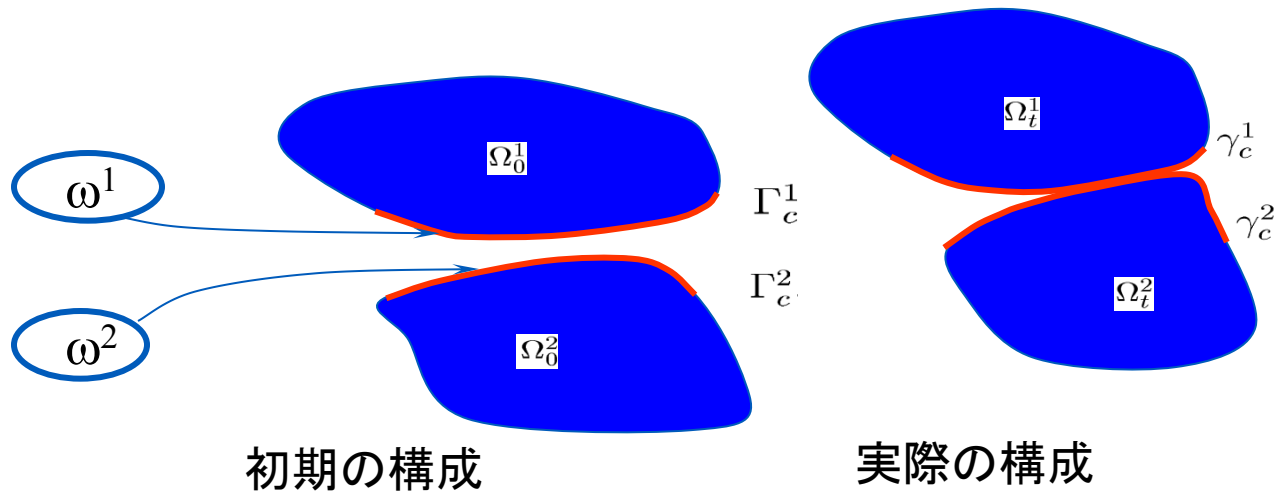
連続体ベースの方法 (1)

▶ 接触摩擦用の新しい方法

- 摩擦接触する変形可能な3D固体
- 大変形
- 準静的または動的な枠組
- 摩擦や剛体運動が多い問題におすすめ

連続体ベースの方法 (2)

▶ 平衡方程式



$$\left\{ \begin{array}{ll} \text{div } \boldsymbol{\Pi}^i &= \mathbf{0} & \text{dans } \Omega_0^i \\ \boldsymbol{\Pi}^i (\mathbb{F}^i)^T &= \mathbb{F}^i (\boldsymbol{\Pi}^i)^T & \\ \boldsymbol{\Pi}^i \mathbf{N}^i &= \mathbf{0} & \text{sur } \Gamma_g^i \\ \boldsymbol{\Pi}^i \mathbf{N}^i &= \mathbf{R}^i & \text{sur } \Gamma_c^i \end{array} \right.$$

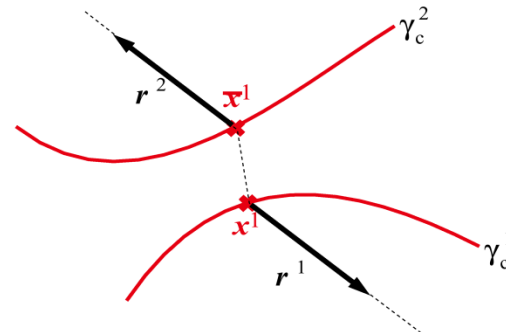
連続体ベースの方法 (3)

▶ 作用と反作用の原理

(PAR: Principle of Action and Reaction)

$$\mathbf{R}^1(\mathbf{p}^1)dS^1 + \mathbf{R}^2(\bar{\mathbf{p}}^1)dS^2 = \mathbf{0}$$

$$\mathbf{r}^1(\mathbf{x}^1)ds^1 + \mathbf{r}^2(\bar{\mathbf{x}}^1)ds^2 = \mathbf{0}$$



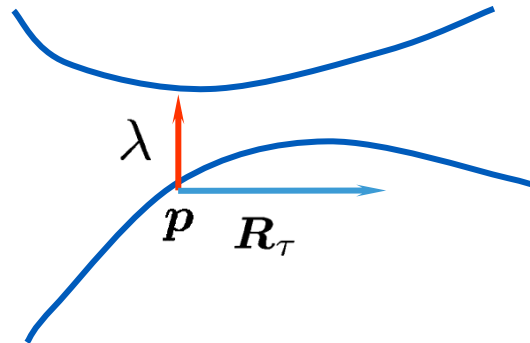
連続体ベースの方法 (4)

▶ インターフェース規則

- 摩擦接触力の各成分への分解

$$\mathbf{R}(\mathbf{p}, t) = \lambda(\mathbf{p}, t)\mathbf{n} + \mathbf{R}_\tau(\mathbf{p}, t)$$

- λ 垂直方向成分 (接触)
- $\mathbf{R}_\tau$ 接線方向成分 (摩擦)

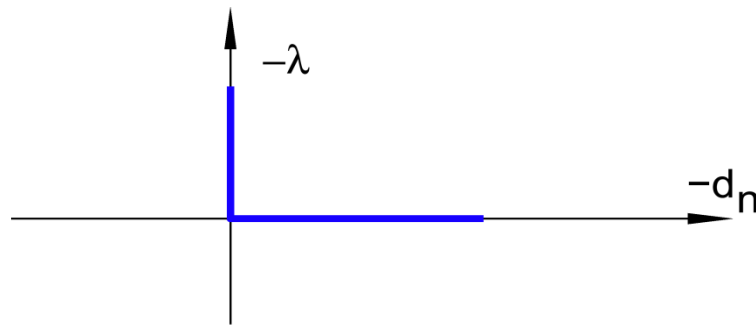
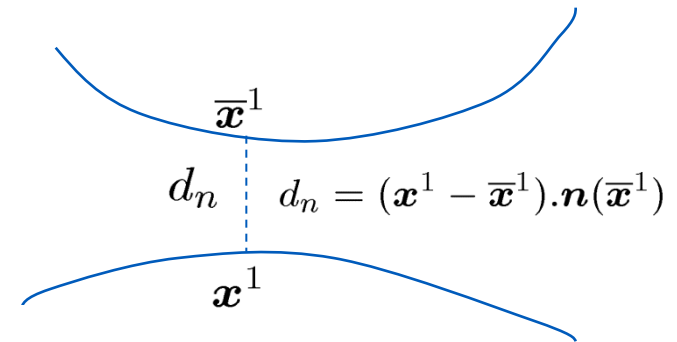


連続体ベースの方法 (5)

▶ 一方向接触 (Hertz-Signorini-Moreau の式)

■ 古典的な表現

$$\begin{cases} d_n(\mathbf{p}, t) & \leq 0 \\ \lambda(\mathbf{p}, t) & \leq 0 \\ \lambda(\mathbf{p}, t) d_n(\mathbf{p}, t) & = 0 \end{cases}$$



連続体ベースの方法 (6)

▶ 一方向接触 (Hertz-Signorini-Moreauの式)

■ 再定式化

$$\lambda(\mathbf{p}, t) - P_{\mathbb{R}_-}(\lambda(\mathbf{p}, t) - \rho_n d_n(\mathbf{p}, t)) = 0 \quad (\rho_n > 0)$$

■ あるいは

$$\lambda - \chi(g_n)g_n = 0 \quad \text{ここで} \quad \begin{cases} g_n = \lambda - \rho_n d_n \\ \chi(x) = \begin{cases} 1, & \text{si } x \leq 0 \\ 0, & \text{si } x > 0 \end{cases} \end{cases}$$

連続体ベースの方法 (7)

▶ クーロン摩擦

■ 古典的な表現

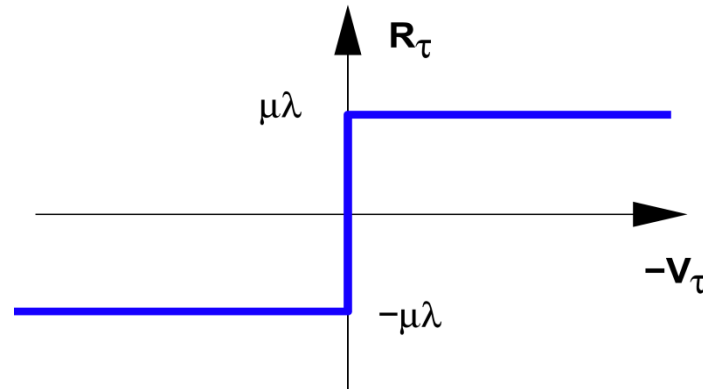
$$\| \mathbf{R}_\tau(\mathbf{p}, t) \| \leq \mu |\lambda(\mathbf{p}, t)|$$

$$\text{Si } \| \mathbf{R}_\tau(\mathbf{p}, t) \| < \mu |\lambda(\mathbf{p}, t)|$$

$$\text{alors } \mathbf{V}_\tau(\mathbf{p}, t) = 0$$

$$\text{Si } \| \mathbf{R}_\tau(\mathbf{p}, t) \| = \mu |\lambda(\mathbf{p}, t)|$$

$$\text{alors } \exists \alpha \geq 0 ; \mathbf{V}_\tau(\mathbf{p}, t) = -\alpha \mathbf{R}_\tau(\mathbf{p}, t)$$



連続体ベースの方法 (8)

▶ クーロン摩擦

■ 再定式化

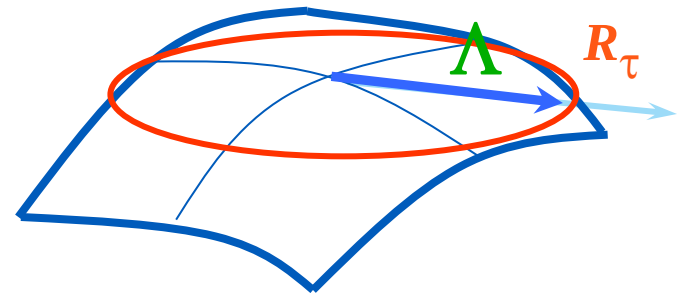
$$\begin{cases} R_\tau(\mathbf{p}, t) = \mu \lambda(\mathbf{p}, t) \Lambda(\mathbf{p}, t) \\ \Lambda(\mathbf{p}, t) \in B(0, 1) \\ \Lambda(\mathbf{p}, t) \cdot \mathbf{V}_\tau(\mathbf{p}, t) = \|\mathbf{V}_\tau(\mathbf{p}, t)\| \end{cases}$$

■ あるいは

$$d_n(\mathbf{p}, t) \Lambda(\mathbf{p}, t) = 0, \quad (\mathbf{p} \in \Gamma_c)$$

$$\begin{cases} R_\tau(\mathbf{p}, t) = \mu \lambda(\mathbf{p}, t) \Lambda(\mathbf{p}, t) \\ \Lambda(\mathbf{p}, t) - P_{B(0,1)}(\mathbf{g}_\tau(\mathbf{p}, t)) = 0 \\ \mathbf{g}_\tau(\mathbf{p}, t) = \Lambda(\mathbf{p}, t) + \rho_\tau \mathbf{V}_\tau(\mathbf{p}, t) \end{cases}$$

$$d_n(\mathbf{p}, t) \Lambda(\mathbf{p}, t) = 0, \quad (\mathbf{p} \in \Gamma_c)$$



連続体に基づく方法 (9)

▶ 3つの場の変数に対する連立方程式

▶ 以下の式から u , λ と Λ を計算 :

- 弱形式の平衡方程式 (仮想仕事の原理)
- 接触条件の弱形式表現
- 摩擦条件の弱形式表現

連続体ベースの方法 (10)

▶ 仮想仕事の原理

$$\begin{aligned} \sum_{i=1}^2 \left\{ \int_{\Omega_0^i} \text{Tr}(\mathbb{F}^i \mathbb{S}^i(\mathbf{u}^i) (\nabla_p(\mathbf{v}^i))^T) d\Omega + \right\} \\ = \int_{\Gamma_c} \chi(g_n) g_n(\mathbf{u}, \lambda) [[\mathbf{v}]]_n d\Gamma + \int_{\Gamma_c} \mu \lambda P_{B(0,1)}(\mathbf{g}_\tau(\mathbf{V}_\tau, \mathbf{\Lambda})) \cdot [[\mathbf{v}]]_\tau d\Gamma \end{aligned}$$

▶ 接触条件の弱形式表現

$$\int_{\Gamma_c} \{ \lambda - \chi(g_n) (\lambda - \rho_n d_n(\mathbf{u})) \} \lambda^* d\Gamma = 0, \quad \forall \lambda^* \in \mathcal{W}(\Gamma_c)$$

▶ 摩擦条件の弱形式表現

$$\int_{\Gamma_c} \chi(g_n) \mu \lambda \{ \mathbf{\Lambda} - P_{B(0,1)}(\mathbf{\Lambda} + \rho_\tau \mathbf{V}_\tau) \} \cdot \mathbf{\Lambda}^* d\Gamma + \int_{\Gamma_c} (\chi(g_n) - 1) \mathbf{\Lambda} \mathbf{\Lambda}^* = 0$$

連続体ベースの方法 (11)

反復解法

- 計算 (既知の繰返回数 n と既知の接触状態 f に対して)

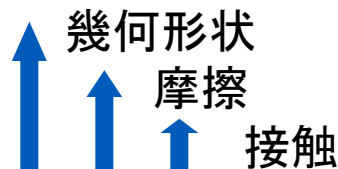
$$\begin{bmatrix} K^n + A_u^f + B_u^{nf} & A^f{}^T & (B^{nf})^T \\ A^f & D^f & 0 \\ B^{nf} & 0 & F^{nf} \end{bmatrix} \begin{Bmatrix} \delta U_h^{n+1} \\ \delta \lambda_h^{n+1} \\ \delta \Lambda_h^{n+1} \end{Bmatrix} = \begin{Bmatrix} L_1^{nf} \\ L_2^{nf} \\ L_3^{nf} \end{Bmatrix}$$

- 接触行列 $[A^f], [A_u^f]$ et $[D^f]$

- 摩擦行列 $[B^{nf}], [B_u^{nf}]$ et $[F^{nf}]$

- 右辺 L_1^{nf}, L_2^{nf} et L_3^{nf}

- ループの反復



連続体ベースの方法 (12)

▶ **FORMULATION = 'CONTINUE'**

▶ いくつかの特別な機能

- 剛体運動を防止するための初期接触を考慮可能
- Dirichlet境界条件と重複する摩擦方向を排除
- さらに詳しい情報を得るには
[U2.04.04] : « 接触要素の使用方法 » 参照のこと

概要

- ▶ 接触の定義
- ▶ 離散の式による、接触問題の解法
 - ペアリング
 - 方程式
 - 解決方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

Code_Aster での使用法 (1)

◆ 接触面の定義: `DEFI_CONTACT`

- キーワード `ZONE` で接触面を定義する

(`ZONE` = 接触する可能性のある面の組み合わせ)

- それぞれの `zone` に対して、接触する可能性のある面は以下で構成 :

- `GROUP_MA_MAIT` : マスター面

- `GROUP_MA_ESCL` : スレーブ面

- 境界の要素がメッシュの中に必ず存在していること (3Dでの2D要素)

- 法線は必ず **外側を向く**

- `ORIE_PEAU_2D`, `ORIE_PEAU_3D` または `ORIE_NORM_COQUE`

- はり(beams)は `ORIE_LIGNE` で接線の向きを指定

- 法線が外側に向いていないと、致命的なエラーとなる

Code_Aster での使用法 (2)

▶ ペアリングのオプション

- 節点と節点のペアリング, マスターとスレーブ, **MAIT_ESCL**
- 節点と節点のペアリング, **NODAL**,
 - **NODAL** 小さな変位を伴うメッシュに互換性がある
- 標準の形式は **MAIT**, **ESCL** および **MAIT_ESCL** である。
 - 法線ベクトルの平均 **MAIT_ESCL** は便利ながある
 - はりの接触に対して **MAIT_VECT_Y** を使って垂直方向を手動定義
 - **DIST_MAIT**, **DIST_ESCL**, **DIST_POUTRE** または **DIST_COQUE** を使えば, さらに隙間 (extra gap) を導入可

Code_Aster での使用法 (3)

▶ ペアリングのオプション

- 法線ベクトルの平滑化 `'NON'` / `'OUI'`
 - 法線ベクトルの平滑化は非常に便利
- 投影の許容値 `TOLE_PROJ_EXT`
 - 投影がマスター面の外側になった時にマスター面を仮想的に延ばす許容値

Code_Aster での使用法 (4)

▶ ペアリングのオプション

- ペアリングにおける ユーザーによる指示 または 自動更新
- オプション **REAC_GEOM**
 - **'AUTOMATIQUE'** : 幾何形状の定常性の基準で自動更新
 - **'CONTROLE'** : 幾何形状の更新回数を、NB_ITER_GEOM で固定値を設定 **注意！**
 - **'SANS'** : 最初の幾何形状を保持 (微小変位). **注意！**

Code_Aster での使用法 (5)

▶ 再計算なしの接触 (**RESOLUTION= 'NON'**)

- 接触を課さずに相互の食い込みを確認するだけ
- 非常に簡易的 (ペアリングのみ)
- き裂の縁の相互食込をチェックするのに便利
- 相互食込を検出するための許容差は **TOLE_INTERP**
- 相互食込が検出された時にオプションを停止させるには **STOP_INTERP**

Code_Aster での使用法 (6)

▶ すべり接触

- アクティブセットまたは連続体に基づく方法によるオプション
`GLISSIERE='OUI'`
- 双方向接触: 相互食込が発生するとすぐに接触が有効になる
- 2つの面の相互のすべりが可能
- 面が分離されそうになった時は接着力を発生させる
(2つの面は分離しない)

Code_Aster での使用法 (7)

▶ Dirichlet境界条件での互換性

- 従属(スレーブ)側の節点にラグランジュ乗数を追加する:
冗長な場合にはゼロピボットが起きる可能性がある
- ラグランジュ乗数を用いた離散の定式化に, 自動排除アルゴリズムが有効
(.mess ファイルの中に “**ZERO PIVOT OTE**” というメッセージがある.)
 - 自動アルゴリズムは、もっぱら 軸に沿って冗長性を検出する
- **SANS_NOEUD** / **SANS_GROUP_NO** によって冗長な節点を設定可能

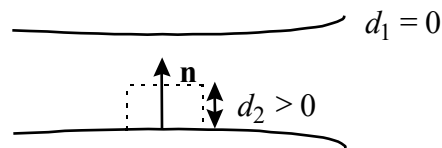
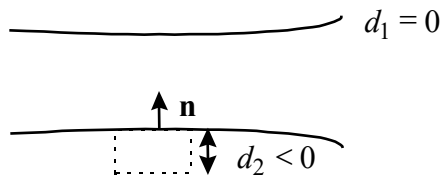
Code_Aster での使用法 (8)

▶ 仮想的な付加的隙間

- オプション `DIST_ESCL` または `DIST_MAINT`

- « 符号付き » 距離

- 負値：何か物体が取り除かれたとき
- 正值：何か物体が追加されたとき



$$(\delta \mathbf{u}_P - \delta \mathbf{u}_M) \cdot \mathbf{N} \leq d - (d_1 + d_2)$$

Code_Aster での使用法 (11)

▶ **STAT_NON_LINE/DYNA_NON_LINE** で 接触が利用可能

- **CONTACT** というキーワードで接触要素を設定
- 離散の接触の定式化で線上での判定は可能だが、この場合の効率はまだ明らかではない
- **STAT_NON_LINE** の **INFO=2** というオプションは、接触に関する(たくさんの)情報を提供する。開発者専用のオプション

Code_Aster での使用法 (12)

▶ 接触のポスト処理

- **CALC_NO**: 節点力 (**FORC_NODA**) と節点の反力 (**REAC_NODA**)を計算
- 接触力 : **POST_RELEVE_T**.で **FORC_NODA** の **RESULTANTE**
節点が運動学的制約に従う場合には要注意
- **STAT_NON_LINE** によって生成されるデータ構造 **VALE_CONT** は, 接触状態についての多くの情報を持っている (隙間, 接触状態, ...)
- 連続体に基づく方法では, **LAGS_C** という自由度は接触圧を指す

概要

- ▶ 接触の定義
- ▶ 離散の式による、接触問題の解法
 - ペアリング
 - 方程式
 - 解決方法
- ▶ 摩擦の定義
- ▶ 摩擦のある接触問題を解くための連続体ベースの式
- ▶ *Code_Aster* の使用法
- ▶ まとめ

チェックリスト

- ▶ ドキュメントを参照のこと!
 - 使用法 [U4.44.11]
 - « *Code_Aster* での接触の使用方法 » [U2.04.04]
 - 理論マニュアル [R5.03.50] と [R5.03.52] (必須ではないが読んだほうが良い)
- ▶ 法線ベクトルは外側に向いているか？
- ▶ 接触を扱う時は行列計算を **STAT_NON_LINE** にしているか？
主に離散の式を使うとき
- ▶ 接触部に冗長なDirichlet境界条件がないか？
- ▶ モデルは OK？ 線形弾性か？

手法の比較

	CPU 計算負荷	Memory 必要な 記憶容量	DOFs 自由度	Robustness ロバスト性
'CONTRAINTE'	++	--	数百	+++
'GCP'	++	+	無制限	+
'PENALISATION'	+	++	数千	-
'CONTINUE'	-	++	無制限	++

End of presentation

Is something missing or unclear in this document?
Or feeling happy to have read such a clear tutorial?

Please, we welcome any feedbacks about Code_Aster training materials.
Do not hesitate to share with us your comments on the Code_Aster forum
[dedicated thread](#).