

高度な非線形解析



Code_Aster, Salome-Meca course material

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

概要

- ▶ 非線形問題の記述
- ▶ 非線形問題解法の理論的背景
- ▶ ニュートン法による非線形問題の解法
- ▶ *Code_Aster*の適用

力学における非線形問題

非線形問題

▶ 非線形問題は以下に由来する：

- 幾何学非線形 (移動とひずみ) : 大変位, 大回転, 大ひずみ
- 材料 : 非線形応答, 履歴依存応答
- 接触/摩擦

▶ 以上3つの非線形性は連成する

▶ 多くのシミュレーションは専門的な解析が必要

非線形材料

▶ 材料非線形

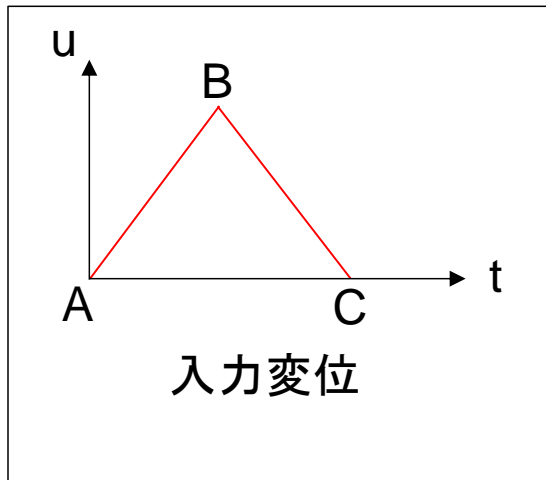
- 実験的な確認：巨視的な力/変位の確認



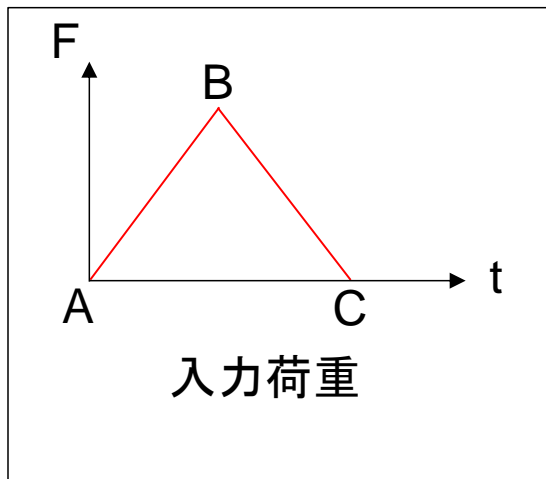
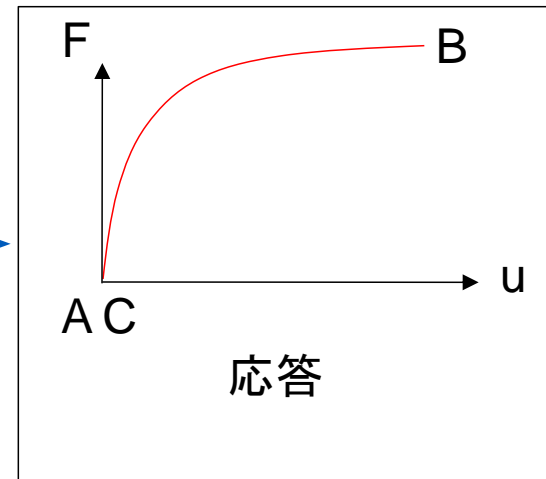
- 多くの材料は非線形 かつ/または 履歴依存

非線形材料

- ▶ 変位/力の関数は線形ではなく、履歴依存ではない



$$F = \varphi(u) \quad \text{with} \quad \varphi(u) \neq a \cdot u + b$$

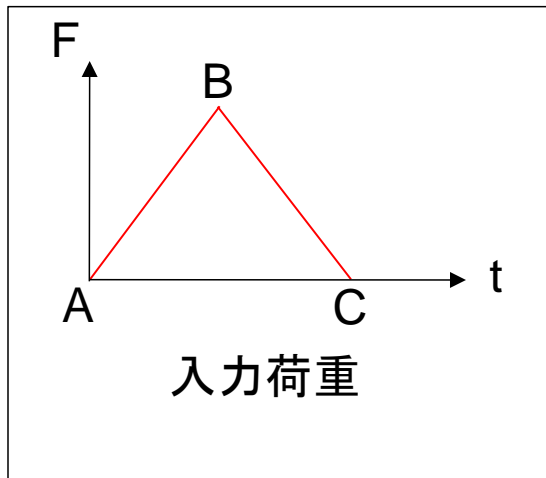
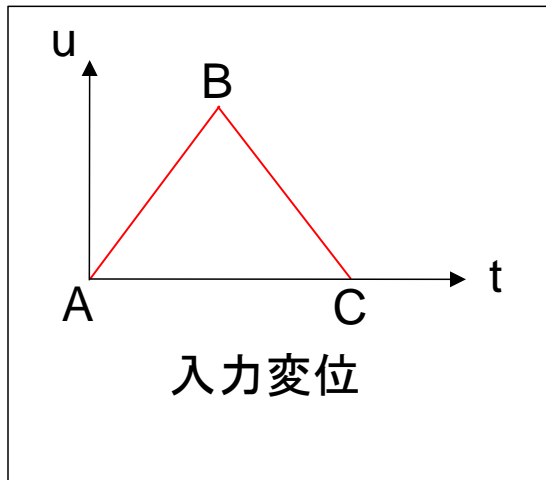


$$u(C) = u(A)$$

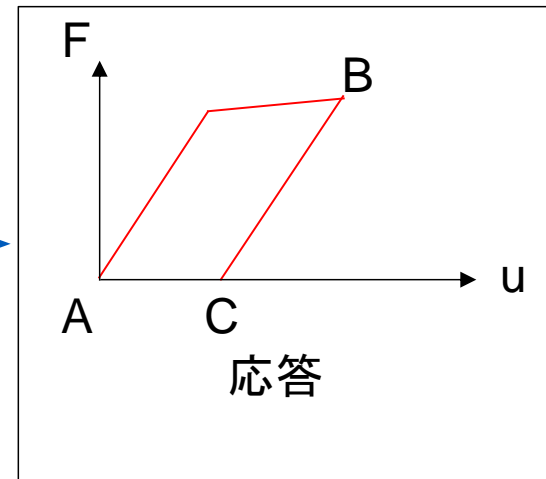
$$F(C) = F(A)$$

非線形材料

- ▶ 変位/力関数は線形ではなくかつ履歴依存



$$F = \varphi(u) \quad \text{with} \quad \varphi(u) \neq a \cdot u + b$$

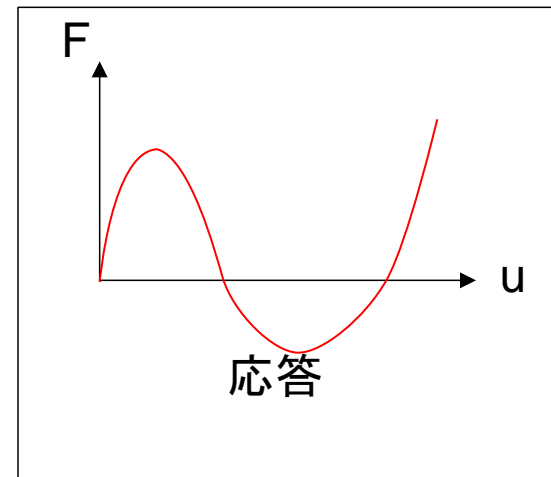
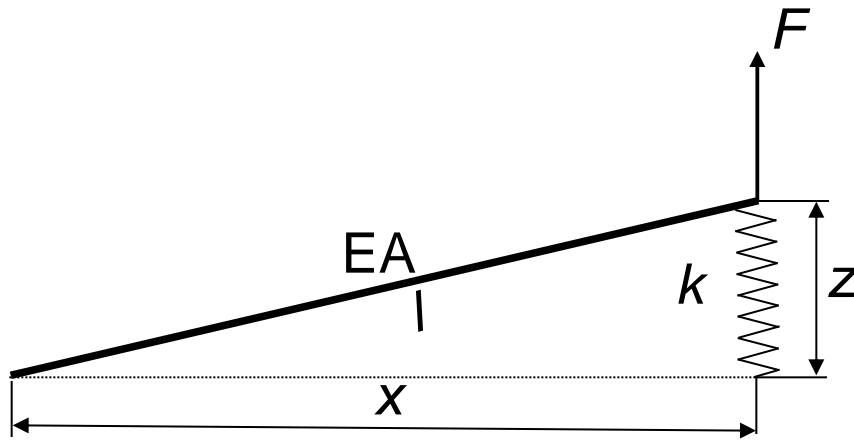


$$u(C) \neq u(A)$$

残差変位
次の荷重：初期
初期変位 $\neq 0$

幾何学非線形

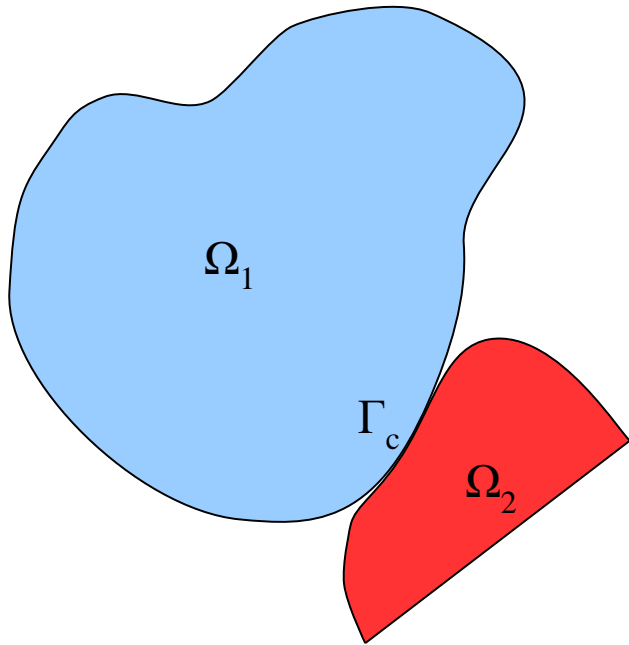
- ▶ 大回転 かつ/または 大変位をともなうトラス/ビーム/シェル要素



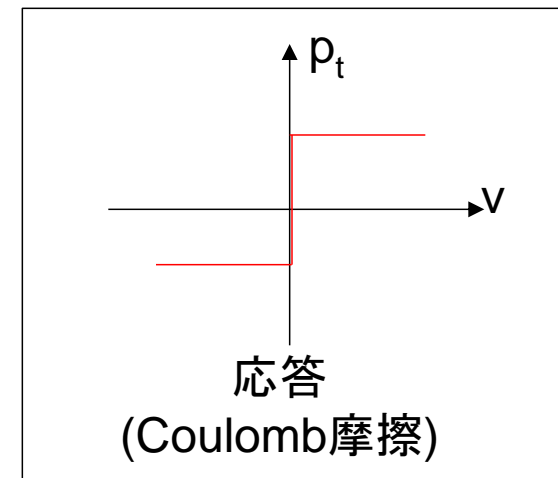
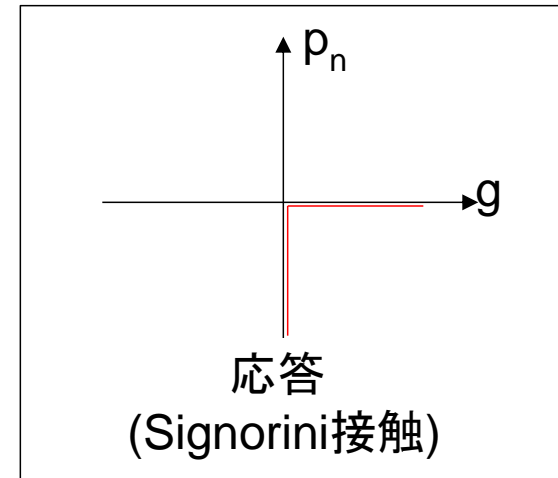
E: ヤング率
A: 断面積
L: 長さ
F: 荷重
u: 変位
k: ばね定数

非線形 接触/摩擦

▶ 接触と摩擦: 非常に難しい非線形問題



- p_n : 垂直な圧力
- p_t : 接線方向の圧力
- g : 物体間の距離 (ギャップ)
- v : 接線方向の物体間の速度

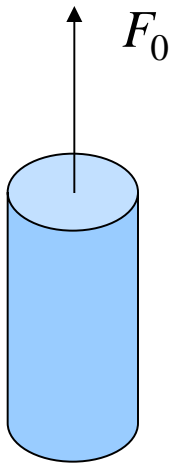


理論的背景

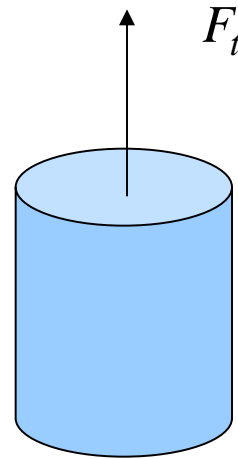
方程式 – 連続体

▶ 応力の測定：コーシー (真) 応力

$$\sigma = \lim_{\Delta S \rightarrow 0} \frac{\Delta F}{\Delta S}$$



$$\sigma_0 = \lim_{\Delta S_0 \rightarrow 0} \frac{\Delta F_0}{\Delta S_0}$$



$$\sigma_t = \lim_{\Delta S_t \rightarrow 0} \frac{\Delta F_t}{\Delta S_t}$$

コーシー応力：変形した時点の配置で測定

▶ ひずみの測定：

- 有限ひずみでは支配法則のもとで任意の選択

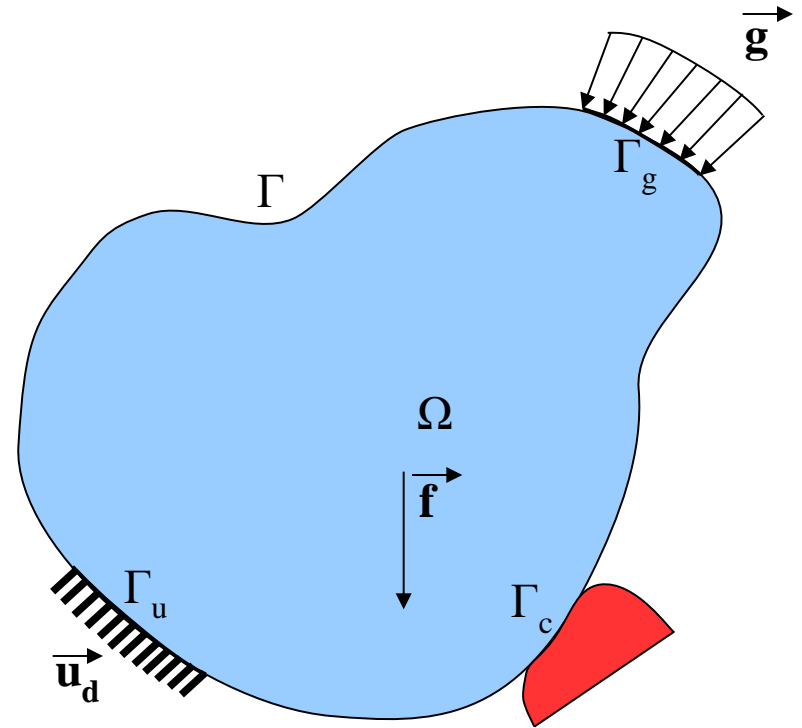
- 微小ひずみでは
$$\varepsilon(u) = \frac{1}{2} (\nabla u + \nabla^t u)$$

方程式 – 連続体

▶ 外力とつり合っている物体 Ω を定義

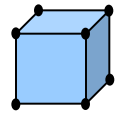
- Γ 外側の境界
- Γ_u 入力変位 \mathbf{u}_d が与えられる境界
- Γ_g 入力荷重 \mathbf{g} が与えられる境界
- Γ_c 接触/摩擦が定義される境界
- \mathbf{f} 体積力

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma} + \mathbf{f} = 0 & \text{in } \Omega \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{g} & \text{on } \Gamma_g \\ \mathbf{u} = \mathbf{u}_d & \text{on } \Gamma_u \\ L_{cf}(\mathbf{u}) & \text{on } \Gamma_c \end{cases}$$

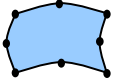


方程式 – 有限要素近似

▶ 主要要素の形 :



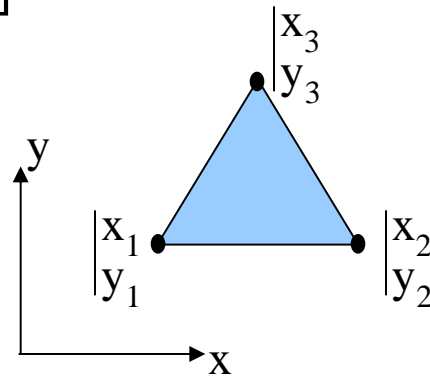
- 面, 三角形, 四角形, 六面体, 四面体, ピラミッド, プリズム



- 線形 または 2次(曲線または直線のエッジ)

▶ 幾何学的離散化 :

- 要素座標系 $\{x\}$
- 節点における離散化された座標系 $\{x_i\}$
- 形状関数 $[N_g]$

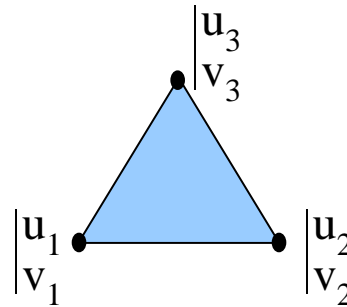


$$\{x\} = [N_g] \{x_i\}$$

方程式 – 有限要素近似

▶ 未知な物理量の離散化:

- 未知量は物理に依存する：
熱解析では熱力学的温度の変位 ...
- 要素内の変位 $\{u\}$
- 離散化された節点変位 $\{u_i\}$
- 形状関数 $[N_u]$



$$\{u\} = [N_u] \cdot \{u_i\}$$

アイソパラメトリック要素を仮定

$$[N] = [N_g] = [N_u]$$

方程式 – 有限要素近似

▶ つりあいの式の弱形式 (重み付き残差) :

- 以下を満たす u を探索 $u \in E_u \quad \forall \tilde{u} \in E_{\tilde{u}}$

$$\int_{\Omega} \sigma(u) : \varepsilon(\tilde{u}) . d\Omega = \int_{\Omega} f(u) : \tilde{u} . d\Omega + \int_{\Gamma_g} g(u) : \tilde{u} . d\Gamma$$

- 仮想変位 \tilde{u} 仮想ひずみ $\varepsilon(\tilde{u}) = \frac{1}{2} (\nabla \tilde{u} + \nabla^t \tilde{u})$

- E_u 空間で運動学的に確認できる変位 :

$$E_u = \{ u \quad \text{with} \quad u(X) = u_d \quad \forall X \in \Gamma_u \}$$

- $E_{\tilde{u}}$ 空間の仮想変位 :

$$E_{\tilde{u}} = \{ \tilde{u} \quad \text{with} \quad \tilde{u}(X) = 0 \quad \forall X \in \Gamma_u \}$$

▶ 弱形式は仮想仕事の原理の概念

- ▶ ガラーキン法 : $\{u\} = [N] \cdot \{u_i\}$ かつ $\{\tilde{u}\} = [N] \cdot \{\tilde{u}_i\}$

方程式 – 有限要素近似

▶ 非線形問題：変位に依存する応力 $\sigma(u)$

- 変位 u からひずみ ε ：幾何学非線形
- ひずみ ε 応力 σ ：材料非線形

▶ 非線形材料

$$\sigma(u) = \sigma(\varepsilon(u))$$

▶ 履歴依存非線形材料 (塑性,...) :

$$\sigma(u) = \sigma(\alpha, \varepsilon(u))$$

内部変数

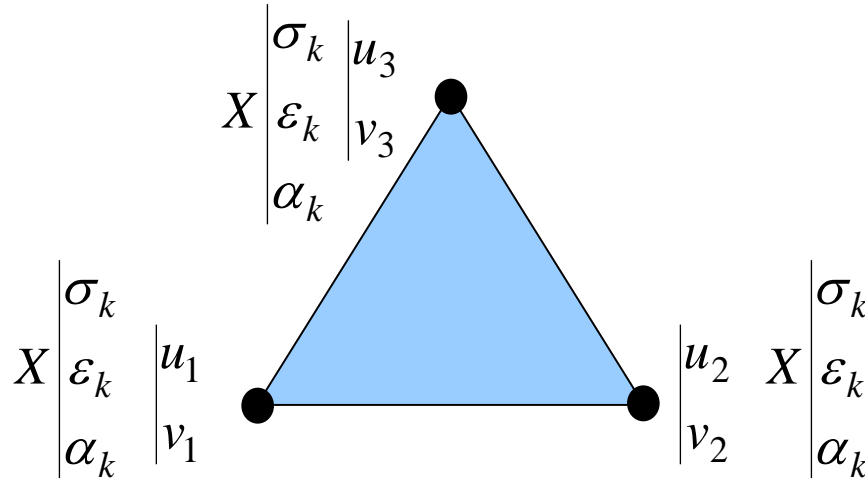
α

方程式 – 有限要素近似

- ▶ 数値的求積法：積分計算にガウス積分を使用

$$\int_{\Omega} \sigma \cdot d\Omega \rightarrow \sum_{k=1}^{\text{nb points}} \sigma_k \cdot \omega_k$$

- ▶ 応力, ひずみ, 内部変数：積分点 X で離散的に定義
(*Code_Aster* では 変数名の後に `_ELGA` がつく量)
- ▶ 変位：節点●において離散化 (*Code_Aster* では 変数名の後に `_NOEU` がつく量)



方程式 – 有限要素近似

▶ 内力の計算

$$\int_{\Omega} \sigma(u) : \varepsilon(\tilde{u}) . d\Omega \rightarrow \{L_{\text{int}}(u)\} = [Q(u)] . \{\sigma(u)\}$$

▶ 非線形内力：変位に依存 u

- 幾何学非線形 $[Q(u)]$
- 材料非線形 $\{\sigma(u)\}$

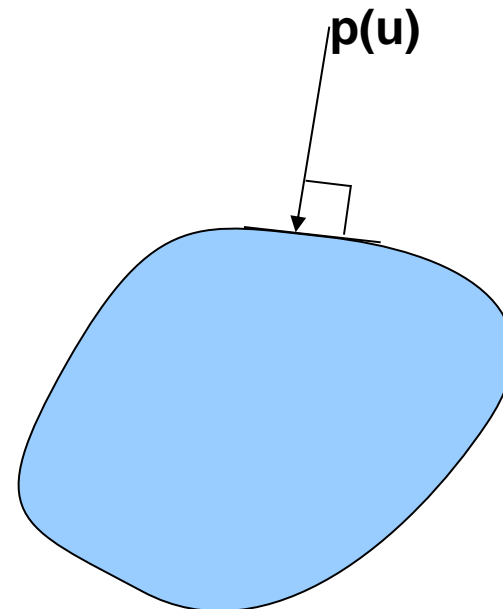
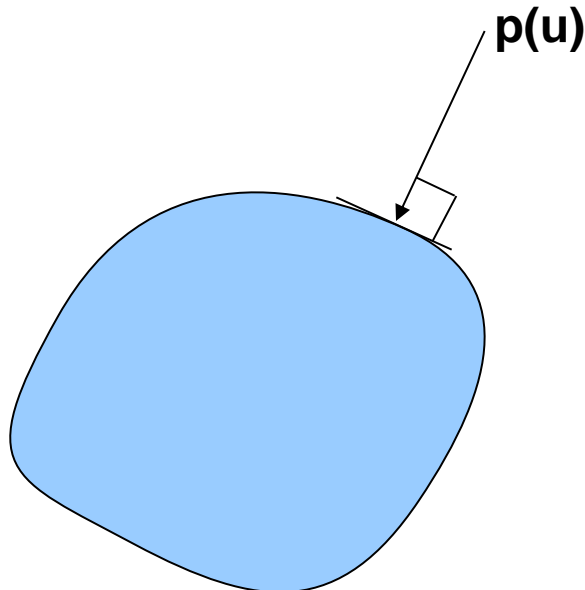
方程式 – 有限要素近似

▶ 外力の計算

$$\int_{\Omega} f(u) : \tilde{u} . d\Omega + \int_{\Gamma_g} g(u) : \tilde{u} . d\Gamma \rightarrow \{L_{ext}(u)\}$$

▶ 非線形外力：変位 u に依存

- 圧力は常に境界に垂直だが → 圧力が変位に依存する場合は、非線形



方程式 – 有限要素近似

▶ 最終的なつり合い方程式：離散化された形

$$\{L_{\text{int}}(u)\} - \{L_{\text{ext}}(u)\} = \{0\}$$

非線形方程式, 変位 u に依存

非線形問題の一般的なアルゴリズム

非線形問題の解法

▶ 非線形問題を解くための：増分アルゴリズム

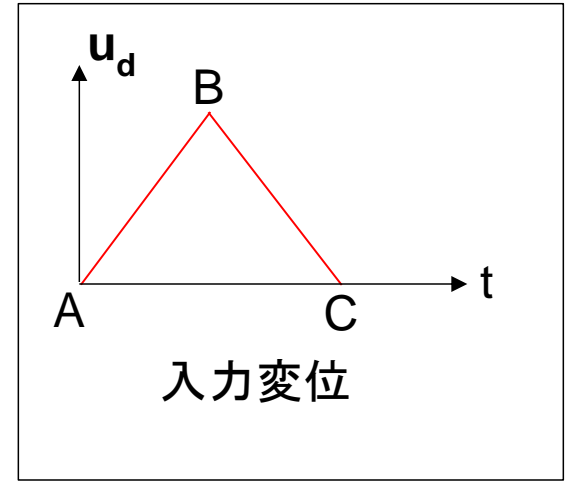
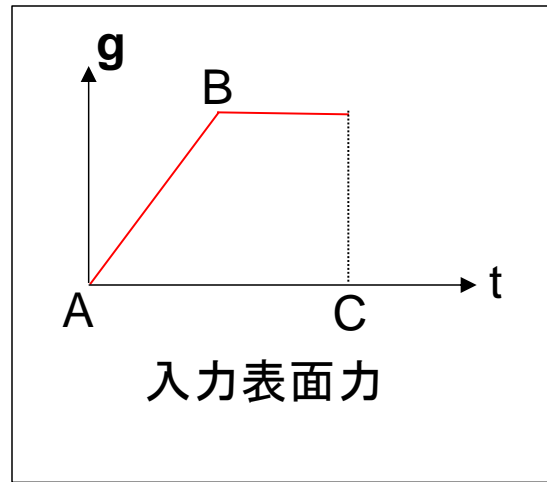
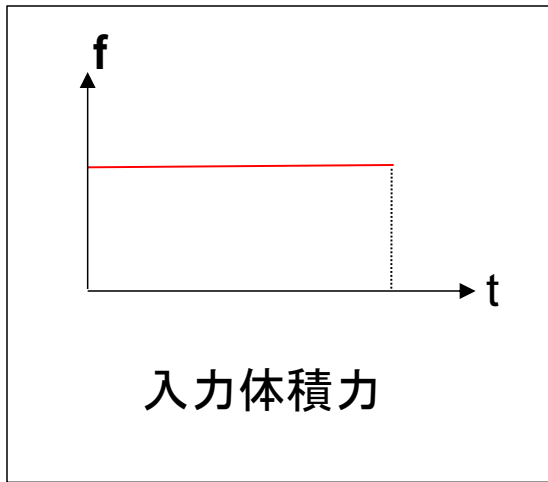
- パラメータ t を使用
- t は実時間ではない (準静的問題)

▶ なぜパラメータか？

- 実際の境界条件には変化する値を使用すべき
- 非線形材料は非線形方程式を生成し、精度が増分に依存
- 小さなステップサイズに区切ることで非線形問題を解きやすくする

非線形問題の解法

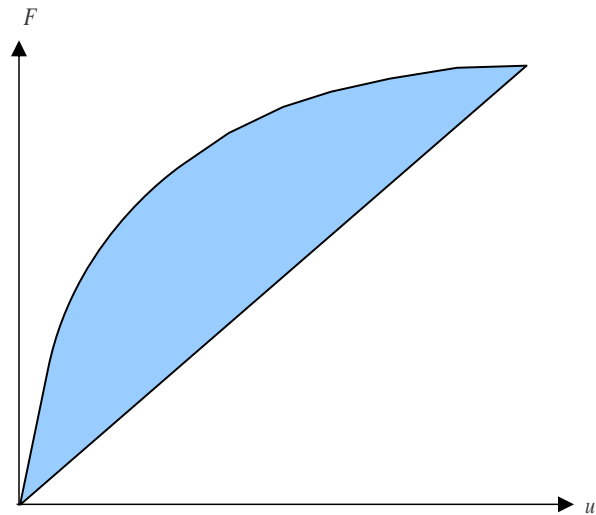
- ▶ パラメータ化: 実際の境界条件には変化する値を適用すべき



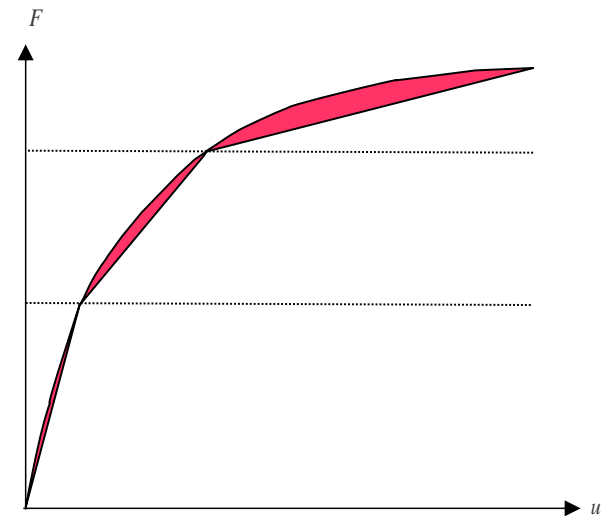
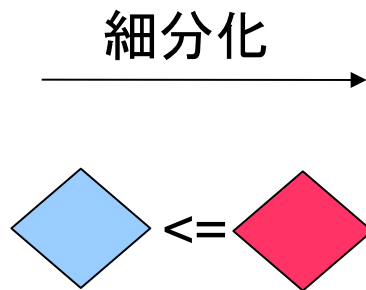
境界条件の例

非線形問題の解法

▶ パラメータ化: 非線形問題を離散化



強い非線形性



個々のステップでは
非線形性が弱くなる

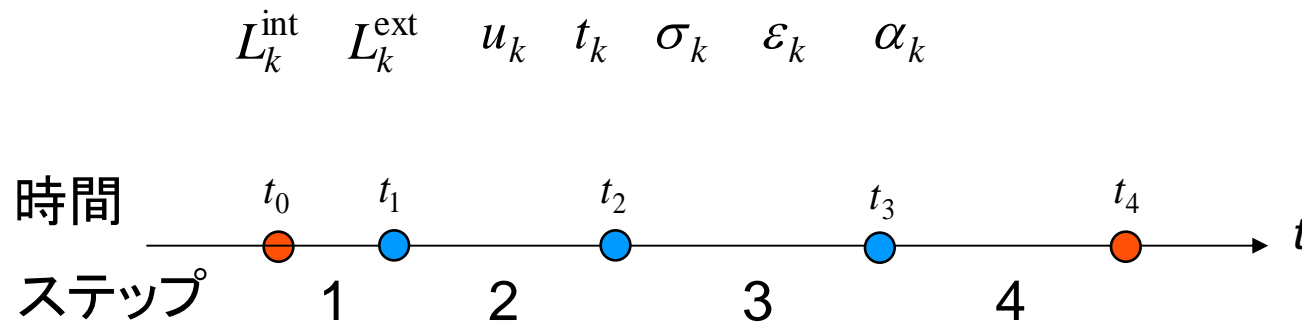
非線形問題の解法

▶ 方程式中のパラメータ t :

- 内力: t に依存するときは, 陰 であり, 構成関係の時間積分の結果得られる
- 外力: t に依存するときは, 陽 (境界条件) または 陰 (追従力)

$$\{L_{\text{int}}(u(t))\} - \{L_{\text{ext}}(t, u(t))\} = \{0\}$$

▶ « 時間 » 離散化, すべての量はステップ k ごとにパラメータ化:



増分: $u_k = u_{k-1} + \Delta u$

非線形問題の解法

▶ 非線形方程式の解法

- 級数の構成
- 1次のテイラー展開

$$F(x) = 0 \quad : \text{ニュートン法}$$
$$(x^n)_{n \in \mathbb{N}}$$

$$0 = F(x^n) \approx F(x^{n-1}) + F'(x^{n-1}) \cdot (x^n - x^{n-1})$$

- 級数の次の値

$$x^n = x^{n-1} - \frac{1}{F'(x^{n-1})} F(x^{n-1})$$

▶ 特徴 :

- 解近傍で2次の収束
- $F'(x^{n-1})$ の計算は高コスト
- $F'(x^{n-1}) \neq 0$

非線形問題の解法

- ▶ ステップ k におけるつりあいの式のためのニュートン法 :

$$\begin{aligned} \{L_{\text{int}}(u_k(t_k))\} - \{L_{\text{ext}}(t_k, u_k(t_k))\} &= \{0\} \\ \{L_{\text{int}}(u(t))\} - \{L_{\text{ext}}(t, u(t))\} &= \{0\} \end{aligned} \quad \begin{array}{c} \downarrow \\ \text{ステップ } k \end{array}$$

- ▶ 内力 :

$$L^{\text{int},n} \approx L^{\text{int},n-1} + \left(\frac{\partial L^{\text{int}}}{\partial u} \right)_{u^{n-1}} \cdot \delta u^n \quad \text{with} \quad \delta u^n = (u^n - u^{n-1})$$

- ▶ 外力 :

$$L^{\text{ext},n} \approx L^{\text{ext},n-1} + \left(\frac{\partial L^{\text{ext}}}{\partial u} \right)_{u^{n-1}} \cdot \delta u^n \quad \text{with} \quad \delta u^n = (u^n - u^{n-1})$$

非線形問題の解法

- ▶ ステップ k におけるつりあいの式のためのニュートン法 :

$$L^{\text{int},n-1} - L^{\text{ext},n-1} + \left(\frac{\partial L^{\text{ext}}}{\partial u} - \frac{\partial L^{\text{int}}}{\partial u} \right)_{u^{n-1}} \cdot \delta u^n = 0$$

$$K^{n-1} \cdot \delta u^n = R^{n-1}$$

- ▶ K^{n-1} は接線剛性マトリクス

$$K^{n-1} = \left(\frac{\partial L^{\text{int}}}{\partial u} \right)_{u^{n-1}} - \left(\frac{\partial L^{\text{ext}}}{\partial u} \right)_{u^{n-1}}$$

- ▶ R^{n-1} はつり合い残差

$$R^{n-1} = \left(L^{\text{ext},n-1} - L^{\text{int},n-1} \right)$$

非線形問題の解法

▶ 全体のアルゴリズム：

- 1) 内力と外力を計算
- 2) 接線剛性マトリクスを計算
- 3) 線形系の求解
- 4) 変位を更新
- 5) 収束判定

非線形問題の解法

▶ 内力と外力の計算：

- 正確な計算が必要 (ソフトウェアの要件)
- 非線形の挙動はガウス点ごとに評価
- 複雑な構成則の計算は高コスト
- 計算精度が変位増分に依存する構成則がある：
 - 粘塑性
 - 半径方向ではない荷重
 - `PETIT_REAC` ひずみ測定

非線形問題の解法

▶ 接線剛性マトリクス計算：

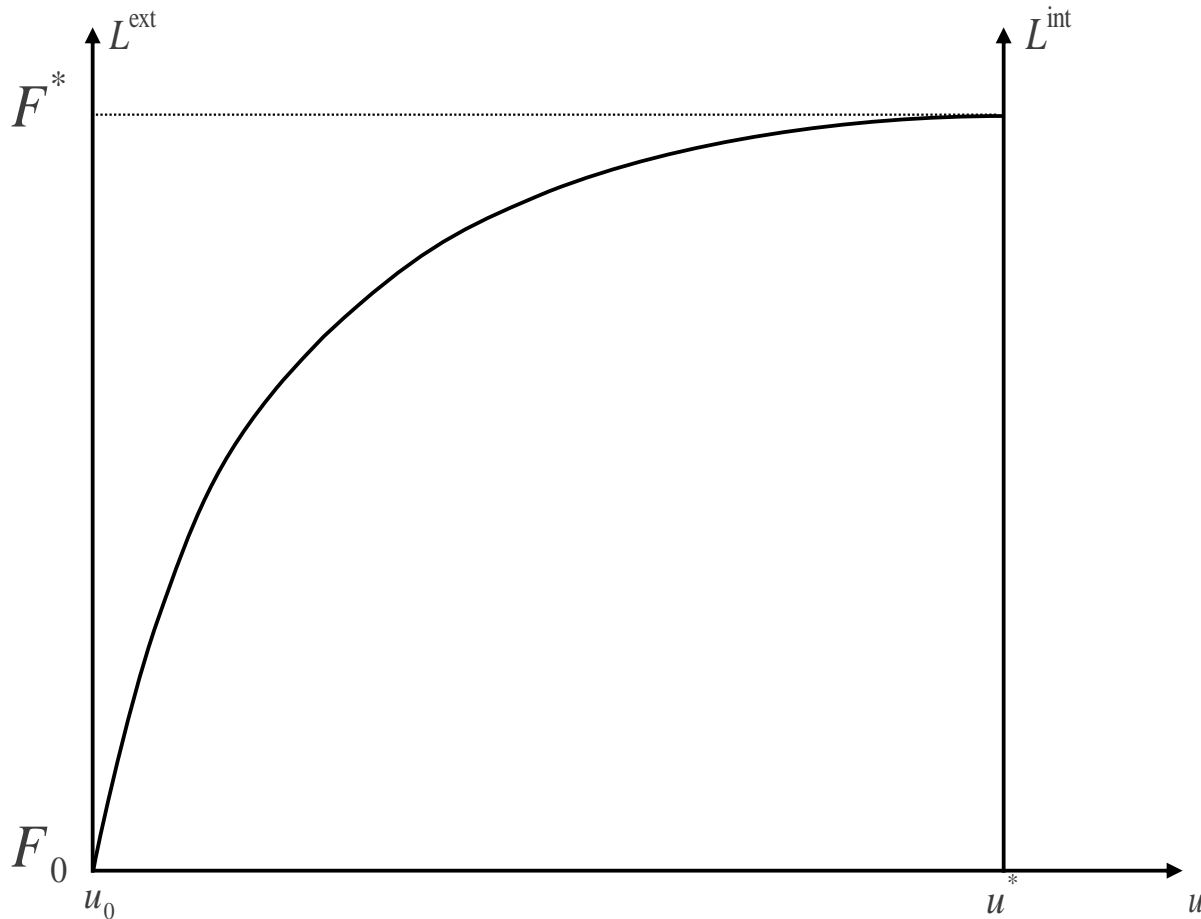
- 正確な計算は必ずしも必要ない
- 接線剛性マトリクス生成時にはガウス点で評価
- 接線剛性マトリクスの計算は高コスト

非線形問題の解法- ニュートン法のグラフ

▶ 解の探索 (u^*, F^*)

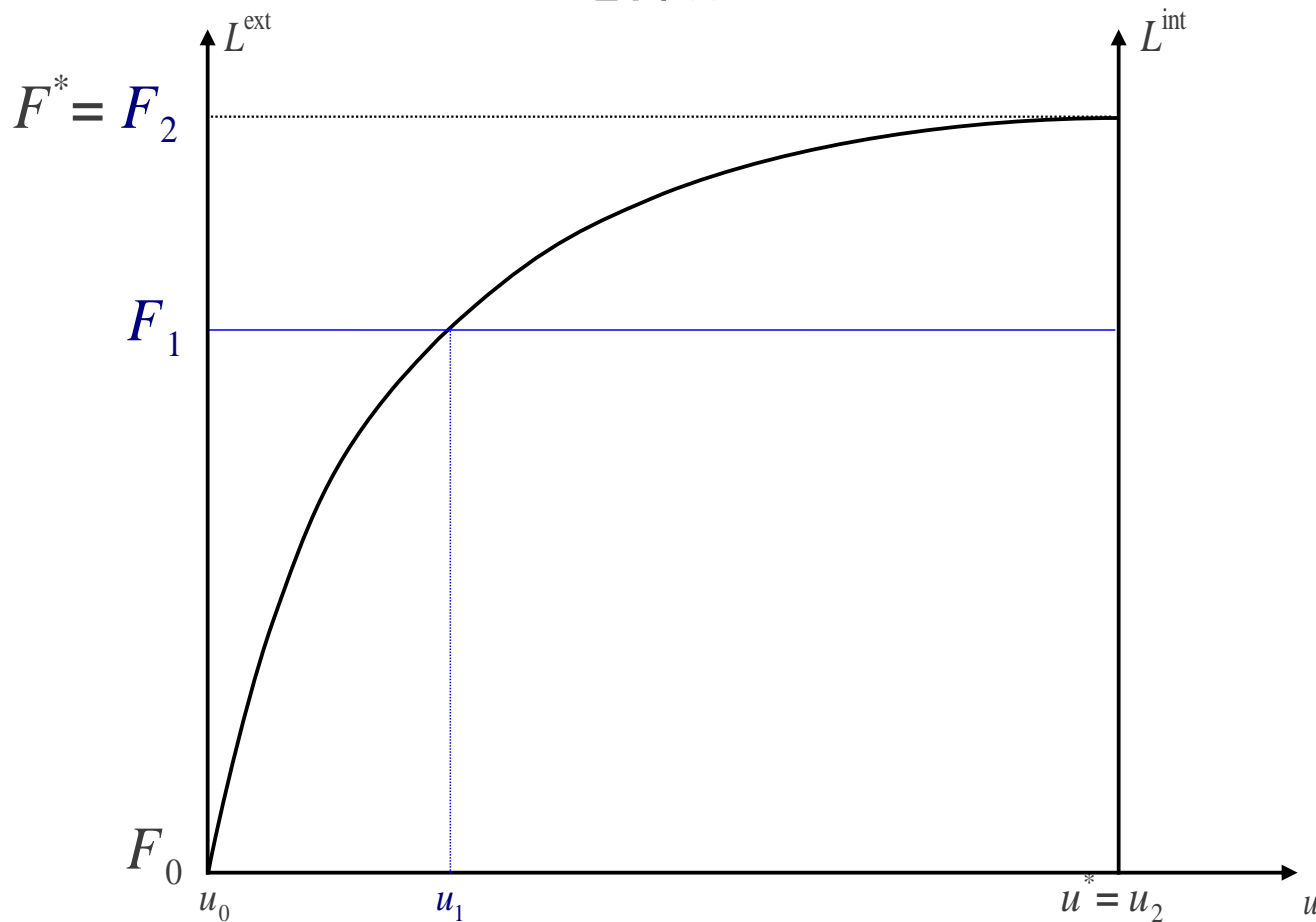
F^* 既知

u^* 未知



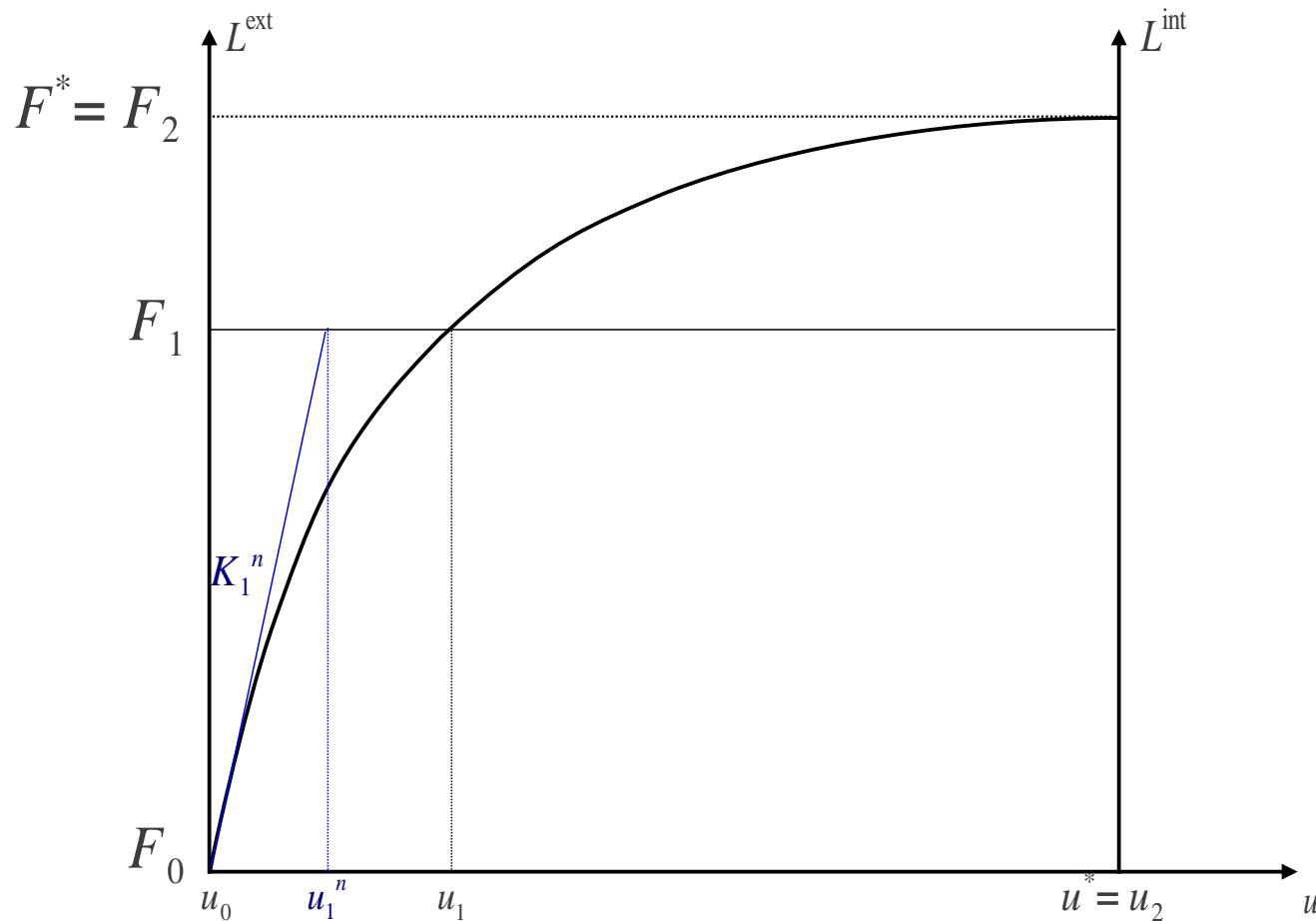
非線形問題の解法—ニュートン法のグラフ

- ▶ F^* を2つの増分に分割 (非線形の度合いを減少) :
 (u_1, F_1) と (u_2, F_2) を探索



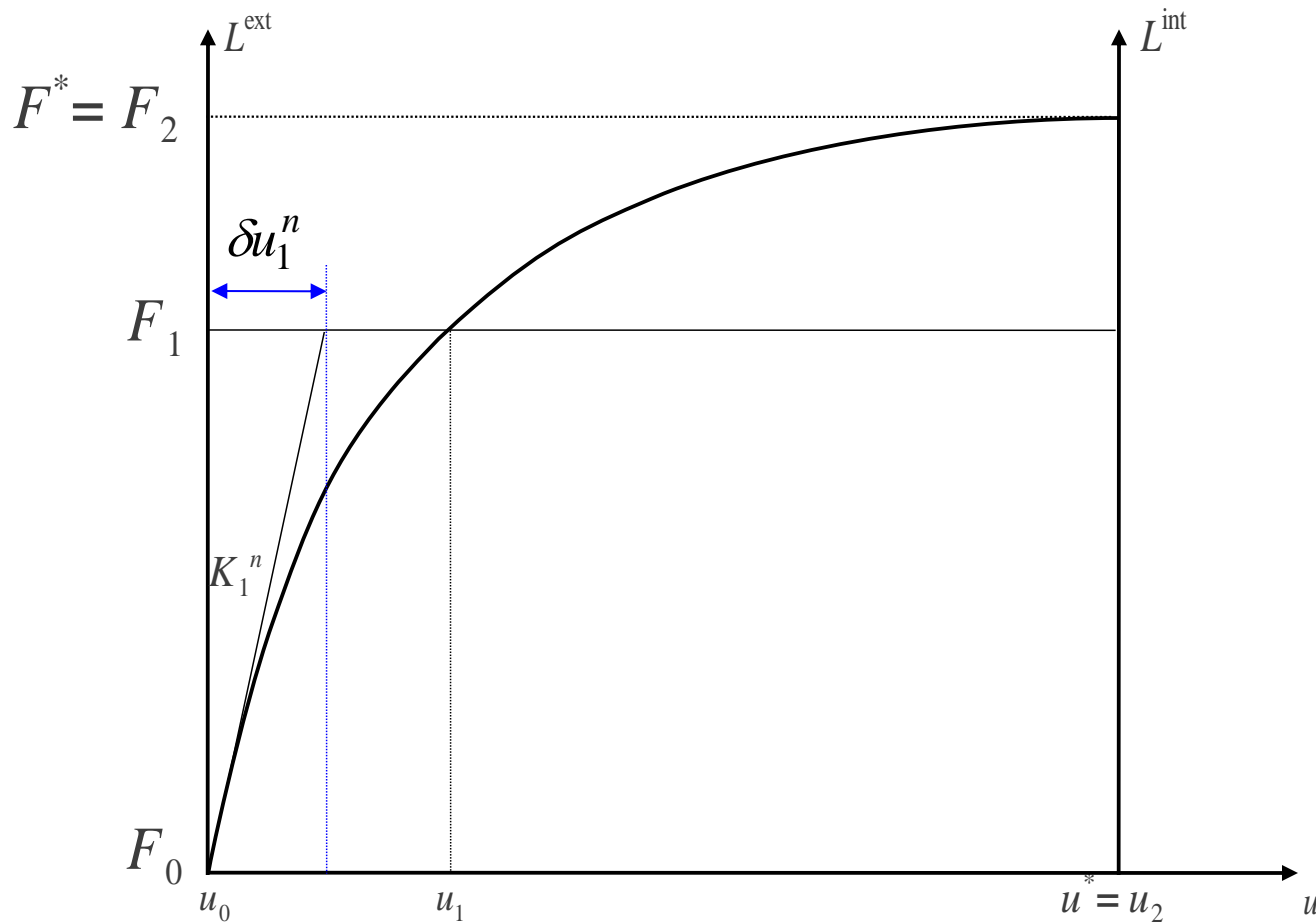
非線形問題の解法—ニュートン法のグラフ

- ▶ n 回の繰り返しに用いる接線剛性マトリクス K_1^n をステップ1で計算



非線形問題の解法- ニュートン法のグラフ

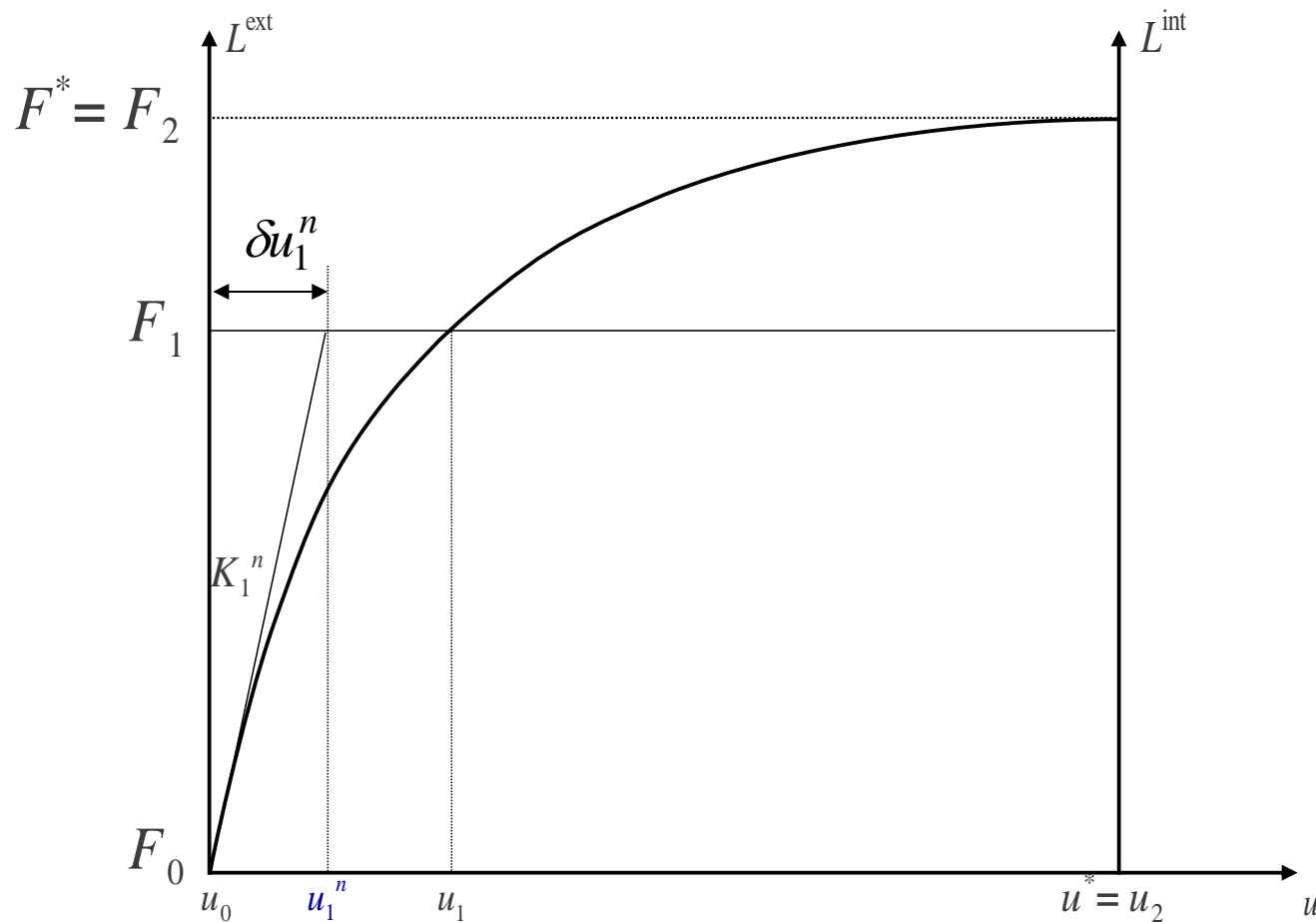
▶ 求解 $K_1^n \cdot \delta u_1^n = F_1 - F_0 \rightarrow \delta u_1^n$



非線形問題の解法- ニュートン法のグラフ

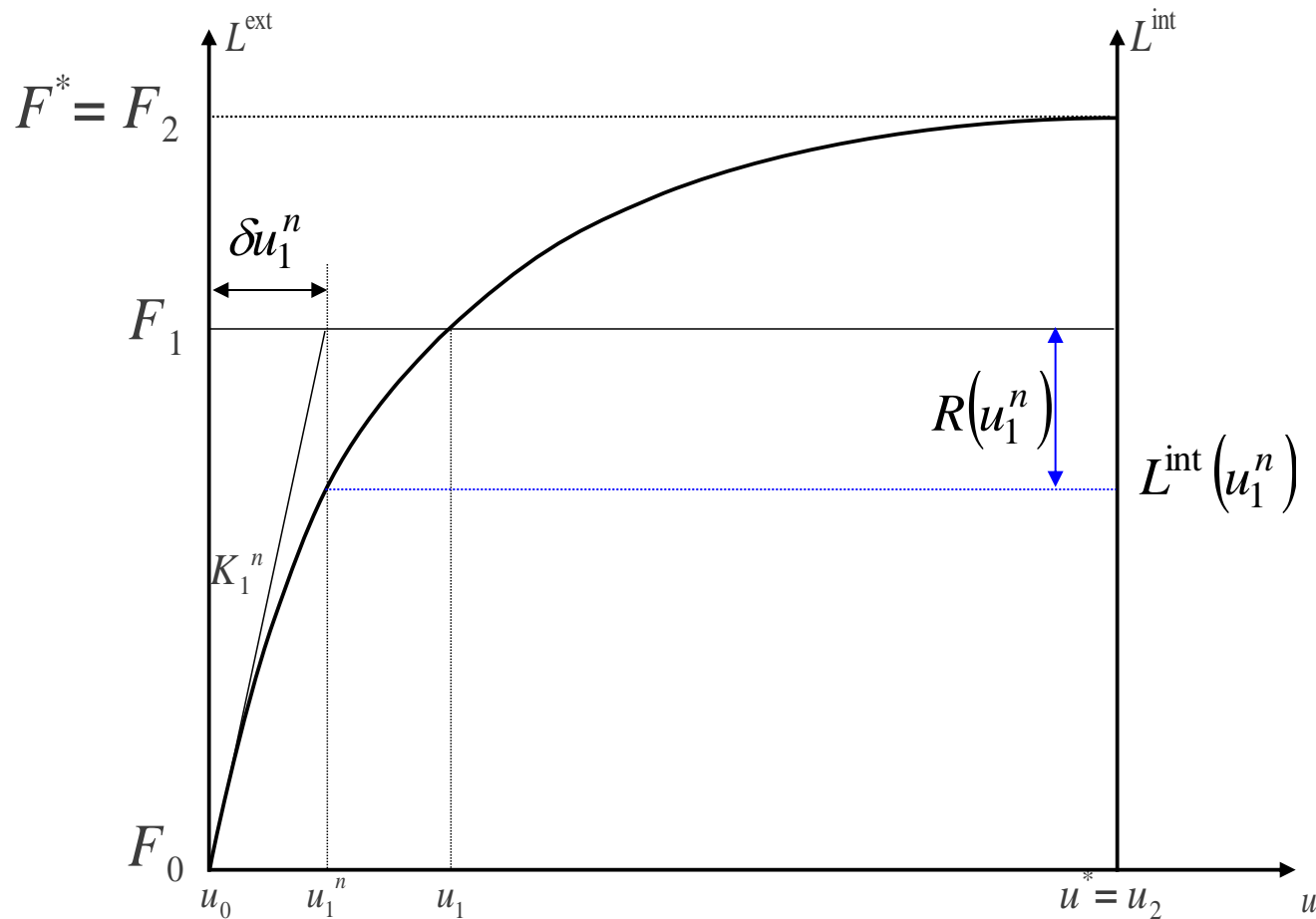
▶ 変位を更新

$$u_1^n = u_1^{n-1} + \delta u_1^n$$

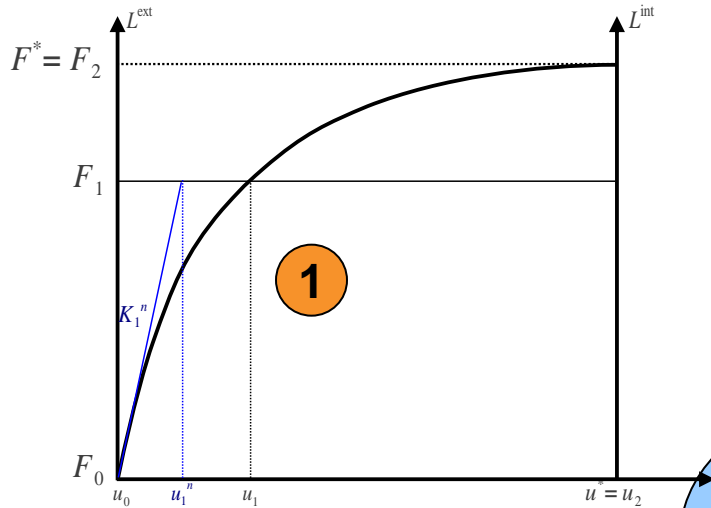


非線形問題の解法- ニュートン法のグラフ

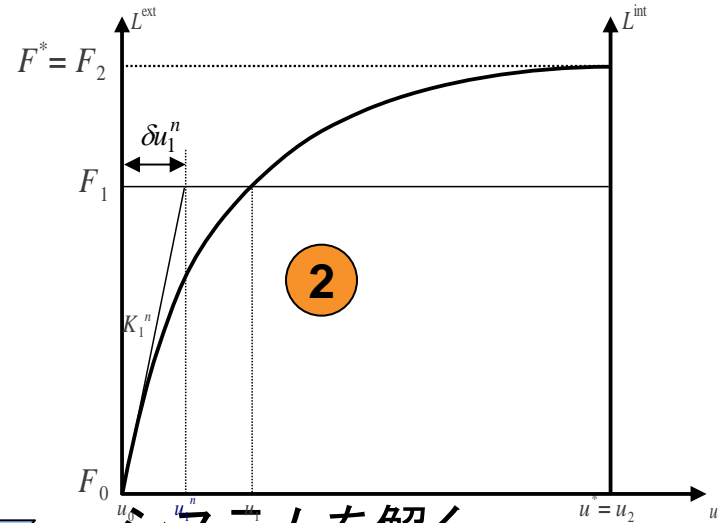
▶ 計算 $L^{\text{int}}(u_1^n)$ と $R = F_1 - L^{\text{int}}(u_1^n)$



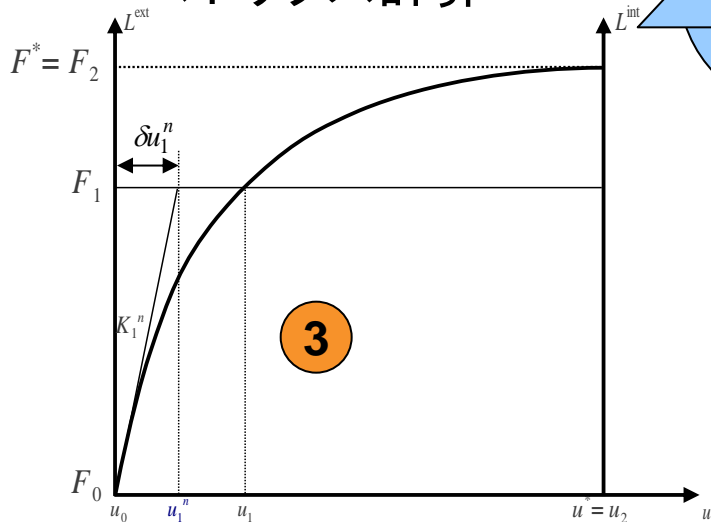
非線形問題の解法- ニュートン法のグラフ



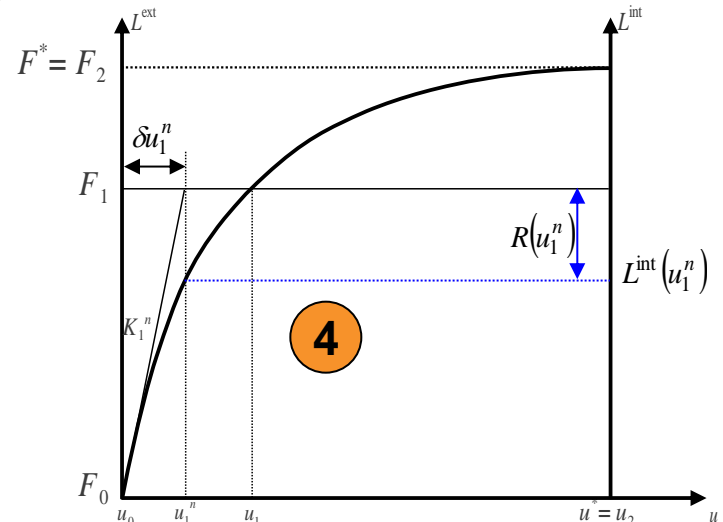
マトリクス計算



システムを解く



変位を更新

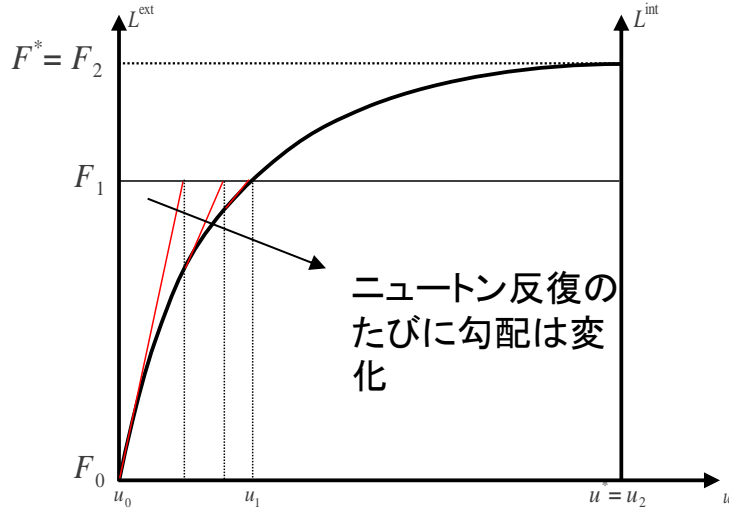


収束判定

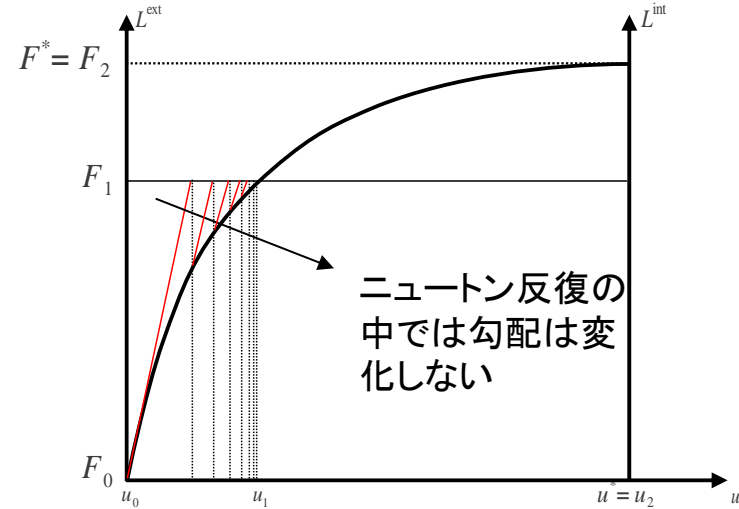
非線形問題の解法

① マトリクス計算：

- 真のニュートン法： K_i^n は全時間ステップ，全Newton反復のたびに評価
- 準ニュートン法： K_i^n は i 番目のNewton反復ごと，全時間ステップ n のたびに評価



真のニュートン法

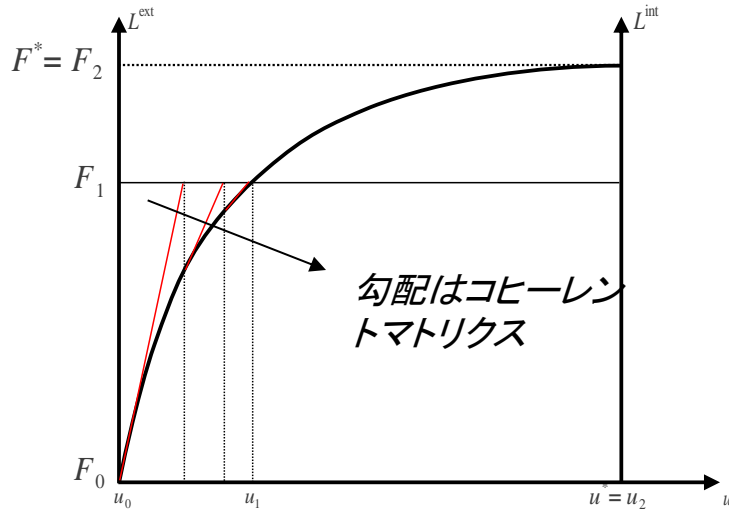


準ニュートン法

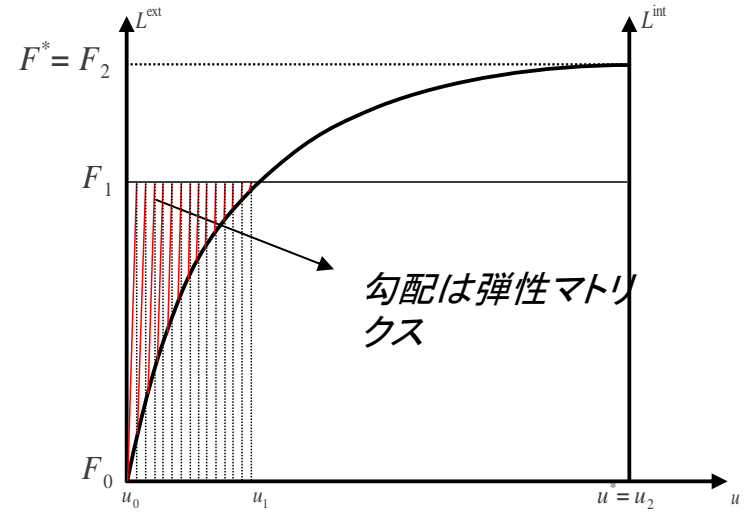
非線形問題の解法

① マトリクス計算：

- 準ニュートン法：概算マトリクス (弾性マトリクス)



真のニュートン法



準ニュートン法

非線形問題の解法

① マトリクス計算：準ニュートン法を使う理由

- すべての繰り返しで正確なマトリクスを計算するには: マトリクスは分解されてなければならない → 非常に高コスト
- 繰り返し回数は増加するが1回の反復計算は速くなる
- 弾性マトリクスはすべての標準的な材料について多数の繰り返しで収束. 弾性マトリクスは1回の計算で計算され分解される: 非常に低コスト
- 弾性マトリクスは, 荷重が除荷の場合にはいい選択

非線形問題の解法

- ② システムを解く：直接解法または反復解法

非線形問題の解法

3 変位を更新 - 直線探索により改良 - 方法

- $K^{n-1} \cdot \delta u^n = R^{n-1}$ を解くことは関数の最小化と等価
- $K^{n-1} \cdot \delta u^n = R^{n-1}$ を解くことは δu^n の方向を与える
- 関数の最小化を解くことは直線探索係数 ρ をあたえる
- 変位を更新

$$u^n = u^{n-1} + \rho \cdot \delta u^n$$

非線形問題の解法

3 変位を更新 - 直線探索により改良 - 関数

- 関数 (スカラ) J は :

$$u \rightarrow J(u) = \int_{\Omega} \Phi(\varepsilon(u)).d\Omega - \int_{\Omega} f.u.d\Omega - \int_{\Gamma} t.u.d\Gamma$$

- $\Phi(\varepsilon(u))$ は超弾性体の自由エネルギー密度 :

$$\varepsilon = \frac{\partial \Phi}{\partial \sigma}$$

- この関数は凸なので, 凸関数を最小化 \rightarrow 勾配がゼロ :

$$\nabla J(u).\tilde{u} = 0 \quad \forall \tilde{u} \in E_{\tilde{u}}$$

- 勾配ゼロ \Leftrightarrow 仮想仕事

$$\int_{\Omega} \sigma(u) : \varepsilon(\tilde{u}).d\Omega = \int_{\Omega} f(u) : \tilde{u}.d\Omega + \int_{\Gamma_g} g(u) : \tilde{u}.d\Gamma \quad \forall \tilde{u} \in E_{\tilde{u}}$$

- つり合いを見つける : J スカラ関数の最小化

非線形問題の解法

3 変位を更新 - 直線探索により改良 - 関数の最小化

- 関数の微分

$$\left\{L_{\text{int}}(u^{n-1} + \rho\delta u^n)\right\} - \left\{L_{\text{ext}}(u^{n-1} + \rho\delta u^n)\right\} = \{0\}$$

- スカラ関数

$$g(\rho) = \left\langle \delta u^n \right\rangle \cdot \left\{L_{\text{int}}(u^{n-1} + \rho\delta u^n)\right\} - \left\{L_{\text{ext}}(u^{n-1} + \rho\delta u^n)\right\} = \{0\}$$

- スカラ凸関数の最小化に古典的な方法を使用 (例えば二分法)

非線形問題の解法

3 変位を更新 - 直線探索により改良 - 注意

- たった数回の繰り返し計算：必ずしも正確に $g=0$ の必要はない
- 内力と外力だけを計算 (マトリクスではない)
- スカラ関数を最小化するアルゴリズムは非常に効率的でシンプル

非線形問題の解法

4 収束判定

- 絶対値による方法 (**RESI_GLOB_MAXI**)

$$\|L^{ext} - L^{int,n-1}\|_{\infty} \leq \zeta$$

- 相対値による方法 (**RESI_GLOB_RELA**)

$$\frac{\|L^{ext} - L^{int,n-1}\|_{\infty}}{\|L^{ext}\|_{\infty}} \leq \zeta$$

- 参照値による方法：応力参照を与える (**RESI_REFE_RELA**)

$$\|L^{ext} - L^{int,n-1}\|_k \leq \gamma \cdot L_k^{ref}$$

非線形へのCode_Asterの適用

非線形へのCode_Asterの適用

- ▶ 材料は等方硬化, Von Mises の降伏条件に従うSTEELの場合, **ファイル**から**曲線データ**を読み込む

```
FSIGM = LIRE_FONCTION (UNITE=21, PARA='EPSI')  
  
STEEL = DEFI_MATERIAU ( ELAS=_F (YOUNG=210.E9, NU=0.3),  
                        TRACTION=_F (SIGM=FSIGM) )
```

- ▶ すべてのメッシュにSTEELを割り当てる

```
CHMA = AFFE_MATERIAU ( MAILLAGE=MESH,  
                       AFFE=_F ( TOUT='OUI',  
                                  MATER='STEEL' ) )
```

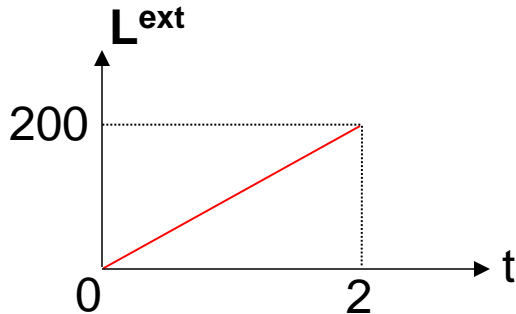
非線形へのCode_Asterの適用

▶ 塑性 等方硬化, 微小ひずみ

```
RESUN = STAT_NON_LINE (...  
    CHAM_MATER = CHMA,  
    COMP_INCR  = _F(  
        TOUT          = 'OUI',  
        RELATION      = 'VMIS_ISOT_TRAC',  
        DEFORMATION  = 'PETIT'),  
    ...)
```

非線形へのCode_Asterの適用

- ◆ 荷重：時間によってパラメータ化 – **FONC_MULT** を使用して関数定義



$$\{L^{ext}(t, u(t))\} = g(t) \cdot \{L^{ext}(u)\}$$

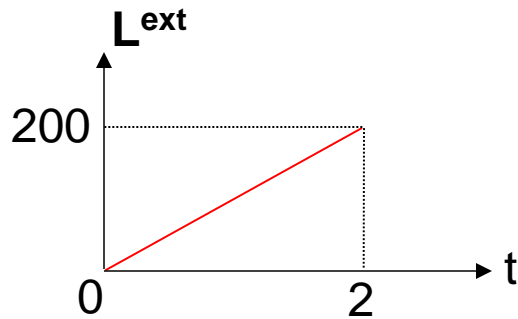
```
RAMPE = DEFI_FONCTION(PARA='INST', VALE=(0,0,2,2))
```

```
LOAD = AFFE_CHAR_MECA(MODELE=MOD,  
                      PRES_REP=_F(PRES=100.,  
                                   GROUP_MA='TOTO'))
```

```
RESUN = STAT_NON_LINE(...  
                      EXCIT=_F(CHARGE =LOAD,  
                               FONC_MULT=RAMPE)  
                      ...)
```

非線形へのCode_Asterの適用

- 荷重：時間によってパラメータ化 – 境界条件で関数を直接定義



$$\{L^{ext}(t, u(t))\} = g(t) \cdot \{L^{ext}(u)\}$$

```
RAMPE = DEFI_FONCTION(PARA='INST', VALE=(0, 0, 2, 200))
```

```
LOAD = AFFE_CHAR_MECA_F(MODELE=MOD,  
                        PRES_REP=_F(PRES=RAMPE,  
                                    GROUP_MA='TOTO'))
```

```
RESUN = STAT_NON_LINE(...  
                      EXCIT=_F(CHARGE =LOAD)  
                      ...)
```

非線形へのCode_Asterの適用

▶ 計算を時間によってパラメータ化

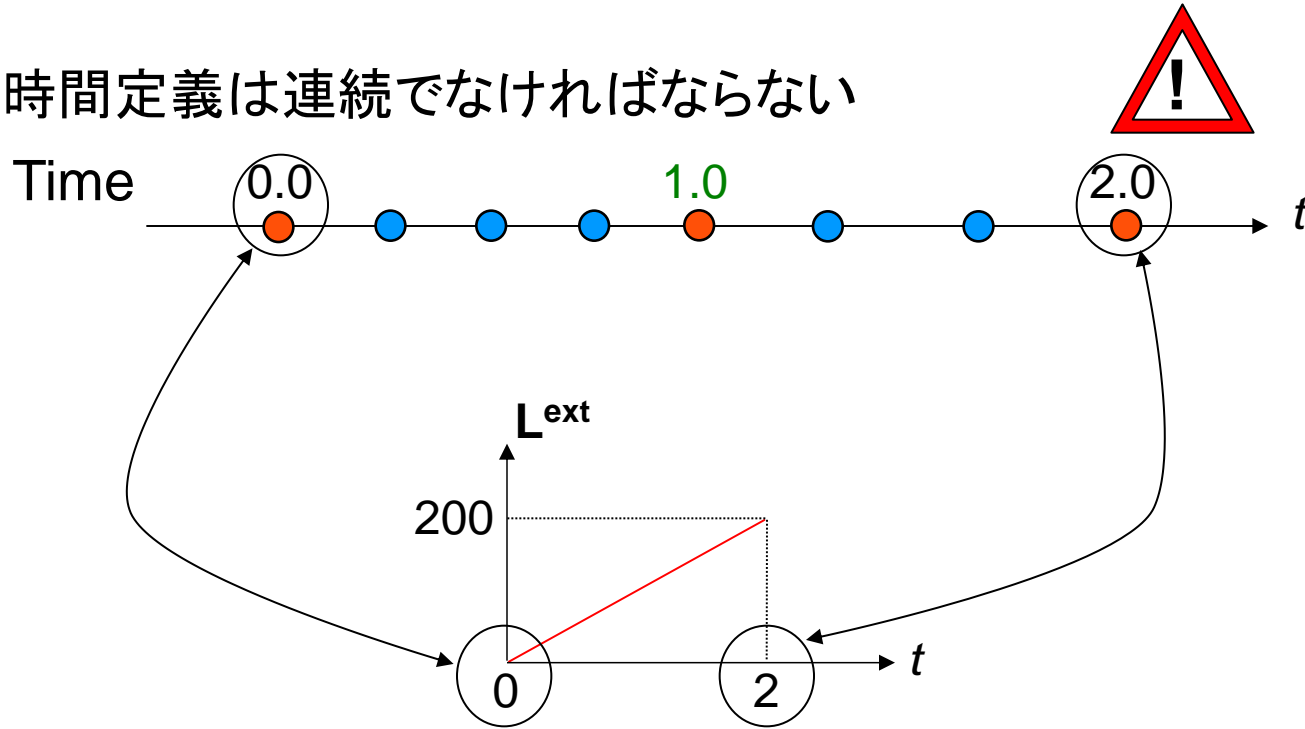


```
L_INST=DEFI_LIST_REEL( DEBUT=0.0,  
                       INTERVALLE=(  
                           _F(JUSQU_A=1.0,NOMBRE=3,)  
                           _F(JUSQU_A=2.0,NOMBRE=2,)  
                       ), )
```

```
RESUN = STAT_NON_LINE(...  
                      INCREMENT=_F(LIST_INST=L_INST)  
                      ...)
```

非線形へのCode_Asterの適用

- ▶ 荷重と時間定義は連続でなければならない



- ▶ ステップ部分での計算

```
RESUN = STAT_NON_LINE (...  
    INCREMENT=_F (LIST_INST=L_INST,  
    INST_FIN =1.0)  
    ...)
```

非線形へのCode_Asterの適用 – 高度な制御



- ▶ 自動ステップ停止：収束しない → 自動停止

```
L_INST=DEFI_LIST_REEL( DEBUT=0.0,  
                       INTERVALLE=(  
                         _F(JUSQU_A=1.0,NOMBRE=3,) ,  
                         _F(JUSQU_A=2.0,NOMBRE=2,) ,  
                       ),)
```

```
DEFLIST = DEFI_LIST_INST(  
           DEFI_LIST = _F(  
             LIST_INST = L_INST,) ,  
           ECHEC      = _F(  
             EVENEMENT = 'ERREUR' ,  
             ACTION     = 'DECOUPE' ,) ,)
```

```
RESUN = STAT_NON_LINE(...  
                      INCREMENT=_F(LIST_INST=DEFLIST)  
                      ...)
```

非線形へのCode_Asterの適用 – 発展的な制御



▶ 自動ステップ適用 :

```
L_INST=DEFI_LIST_REEL( DEBUT=0.0,  
    INTERVALLE=( _F(JUSQU_A=2.0,NOMBRE=1,) ,  
    ) , )
```

```
DEFLIST = DEFI_LIST_INST( DEFI_LIST = _F(  
    METHODE      = 'AUTO`'  
    LIST_INST    = L_INST ,  
    PAS_MINI     = 1.E-6) , )
```

```
RESUN = STAT_NON_LINE (...  
    INCREMENT=_F(LIST_INST=DEFLIST)  
    ...)
```


非線形へのCode_Asterの適用 – 発展的な制御

- ▶ 準ニュートン法 : Newton 反復 2 回ごとに再計算

```
RESUN = STAT_NON_LINE (...  
                NEWTON=_F (REAC_ITER=2, ), )
```

- ▶ 準ニュートン法 : Newton 反復 3 回ごとに再計算

```
RESUN = STAT_NON_LINE (...  
                NEWTON=_F (REAC_INCR=3, ), )
```

- ▶ 準ニュートン法 : 弾性マトリックスを使用

```
RESUN = STAT_NON_LINE (...  
                NEWTON=_F (MATRICE='ELASTIQUE', ), )
```

- ▶ Code_Asterのデフォルト値

```
RESUN = STAT_NON_LINE (...  
                NEWTON=_F ( REAC_INCR=1,  
                          REAC_ITER=0,  
                          MATRICE='ELASTIQUE', ), )
```



非線形へのCode_Asterの適用 – 高度な制御

- ▶ 収束の判定値：値を変えるときは注意！



```
RESUN = STAT_NON_LINE (...  
                CONVERGENCE=_F(  
                RESI_GLOB_MAXI=1.E-6,  
                RESI_GLOB_RELA=1.E-6,))
```

- ▶ 収束の判定値の値を変えるとまちがった結果となりうる
- ▶ ニュートン法の最大繰返し回数



```
RESUN = STAT_NON_LINE (  
                CONVERGENCE=_F(ITER_GLOB_MAXI=20,))
```

- ▶ 弾性マトリクスは非常に多くの繰返しが必要
- ▶ Code_Asterのデフォルト値

```
RESUN = STAT_NON_LINE (...  
                CONVERGENCE=_F(  
                ITER_GLOB_MAXI=10,  
                RESI_GLOB_RELA=1.E-6,))
```

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](#).