Chapter 3
Nonlinear Finite Element Analysis

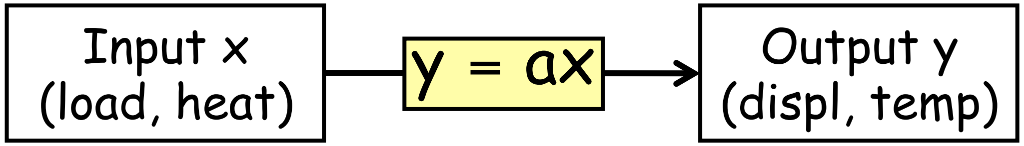
Introduction

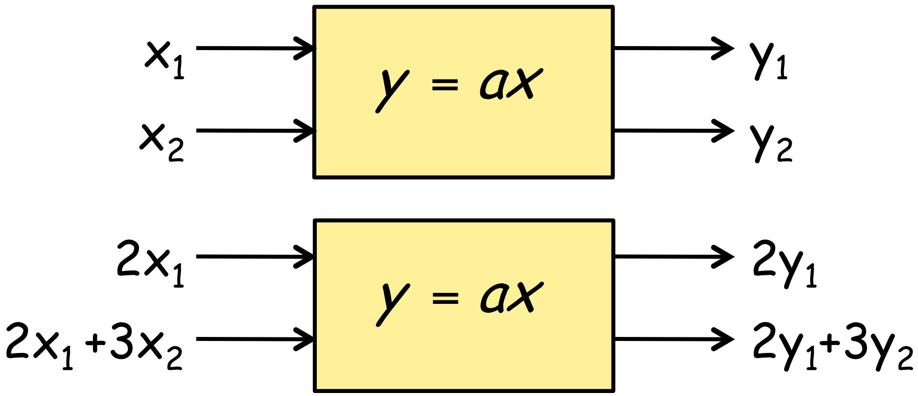
Nonlinear Structural Problems

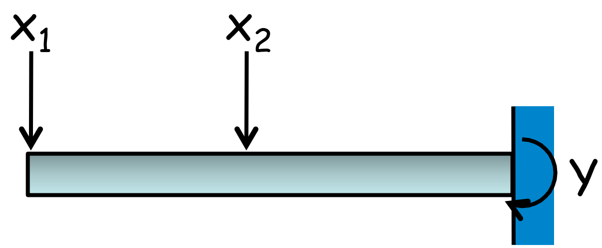
- What is a nonlinear structural problem?
  - Everything except for linear structural problems
  - Need to understand linear problems first

- What is linearity?

$$A(\alpha u + \beta w) = \alpha A(u) + \beta A(w)$$

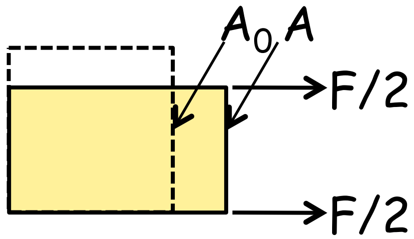




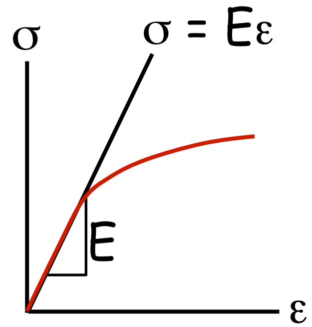


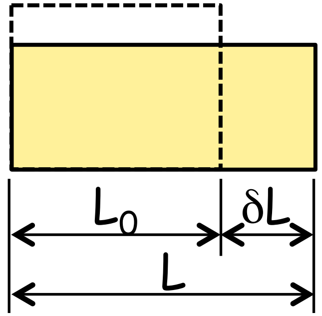
What is a linear structural problem?

Global		Local		Local		Global
Force	Linear →	Stress	Linear →	Strain	Linear →	Displacement



$$\sigma_{nom} = \frac{F}{A_0}, \sigma_{inst} = \frac{F}{A(F)}$$





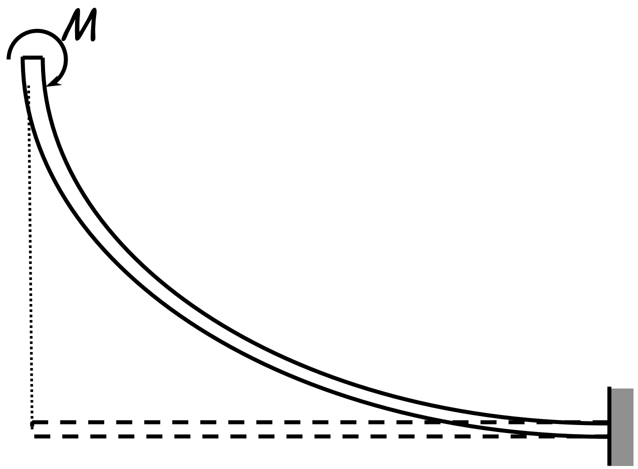
$$\varepsilon_{nom} = \frac{\delta L}{L_0}, \varepsilon_{inst} = \frac{\delta L}{L}$$

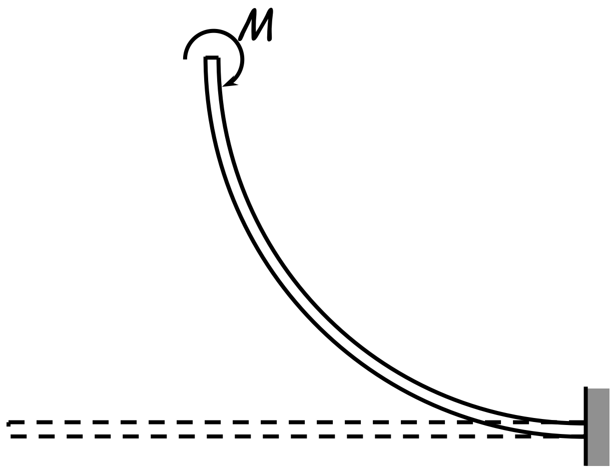
- Linearity is an approximation
- Assumptions:
  - Infinitesimal strain (<0.2%)
  - Infinitesimal displacement
  - Small rotation
  - Linear stress-strain relation

$$F = \sigma A_0 = A_0 E\varepsilon = \frac{A_0 E}{L_0}\delta L$$

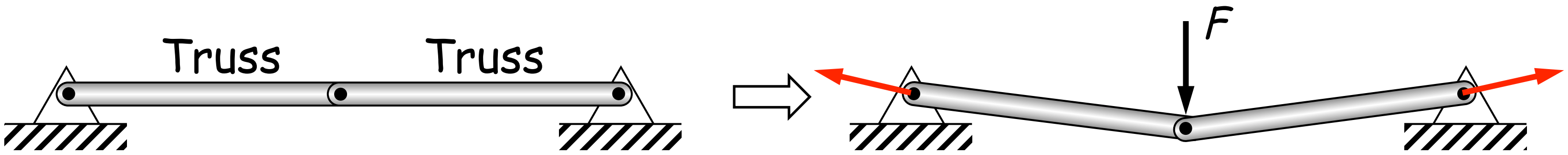
Observations in linear problems

- Which one will happen?





- Will this happen?



What types of nonlinearity exist?

Geometrical

Material

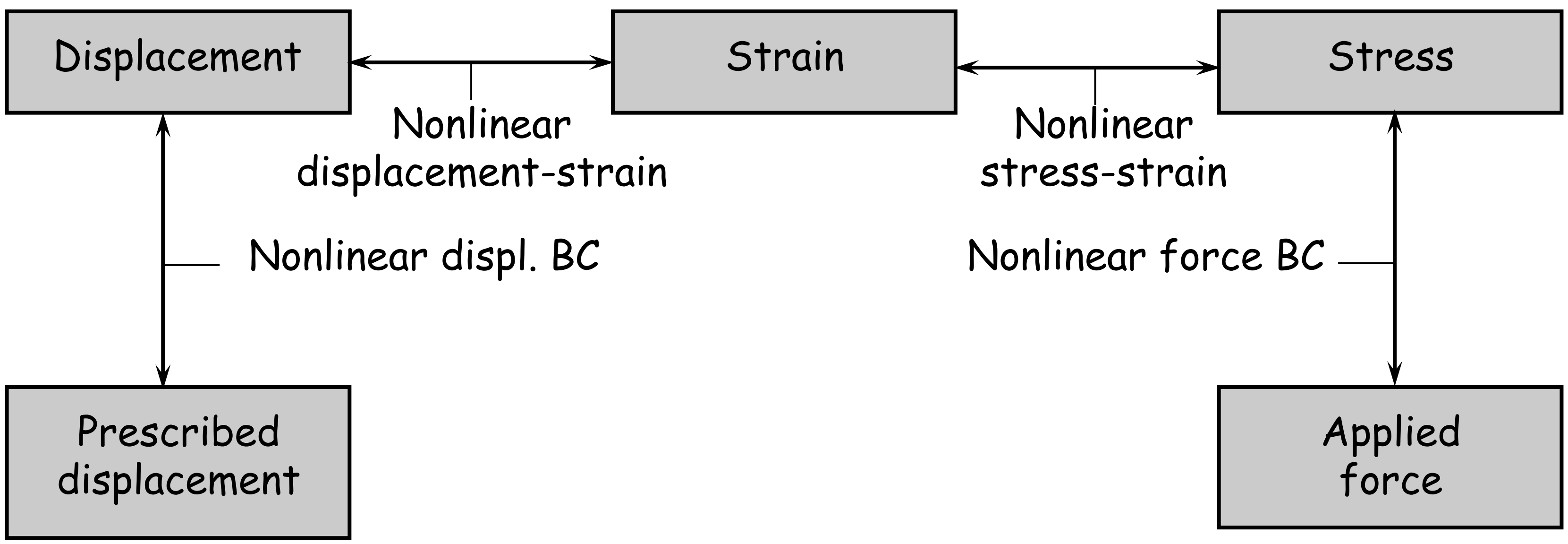
Through BCs

It can be at every stage of analysis!

Linear vs. Nonlinear Problems

- Linear Problem:
  - Infinitesimal deformation: $\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ Undeformed coord.
  - Linear stress-strain relation: $\sigma = D : \varepsilon$ Constant
  - Constant displacement BCs
  - Constant applied forces
- Nonlinear Problem:
  - Everything except for linear problems!
  - **Geometric nonlinearity**: nonlinear strain-displacement relation
  - **Material nonlinearity**: nonlinear constitutive relation
  - **Kinematic nonlinearity**: Non-constant displacement BCs, contact
  - **Force nonlinearity**: follow-up loads

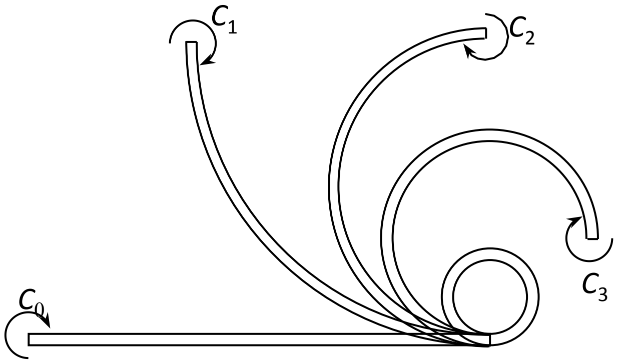
Nonlinearities in Structural Problems

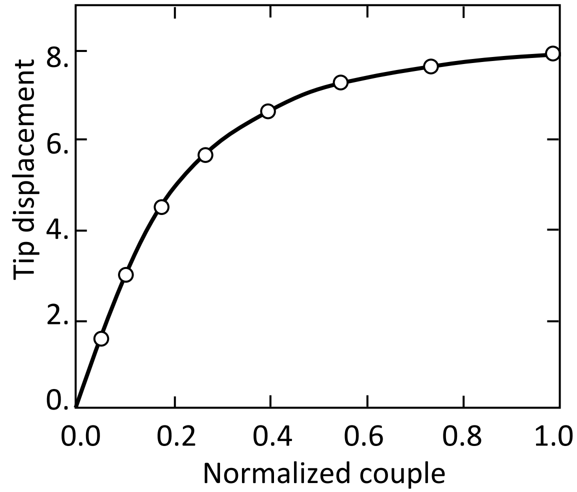


- **More than one nonlinearity can exist at the same time**

Geometric Nonlinearity

- Relations among kinematic quantities (i.e., displacement, rotation and strains) are nonlinear





- Displacement-strain relation
  - Linear: $\varepsilon(x) = \dfrac{du}{dx}$
  - Nonlinear: $E(x) = \dfrac{du}{dx} + \dfrac{1}{2}\left(\dfrac{du}{dx}\right)^2$

When du/dx is small

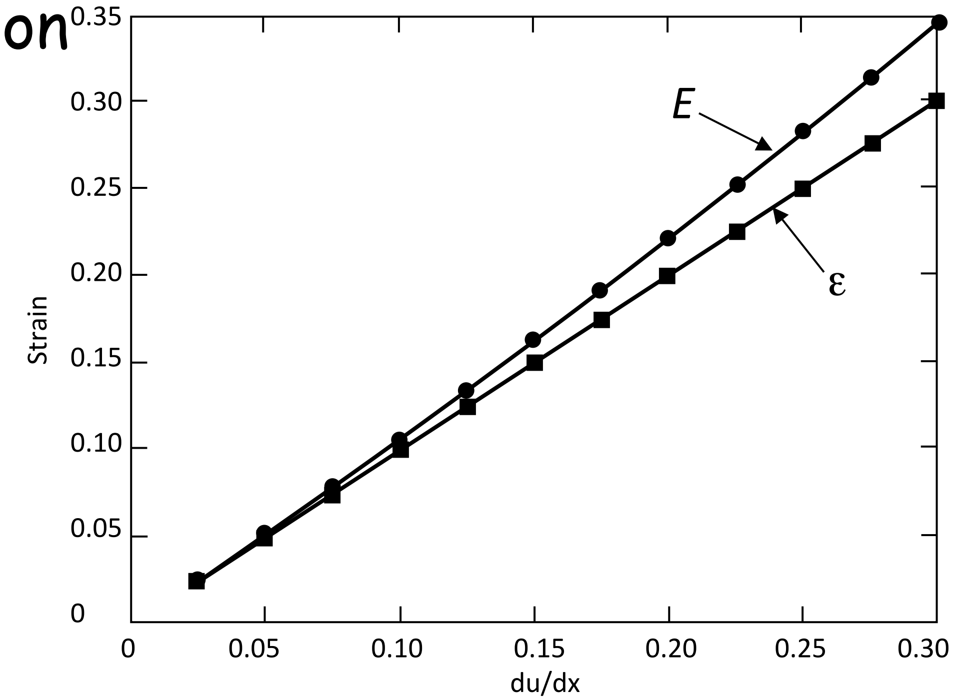
$$\left(\frac{du}{dx}\right)^2 \ll \frac{du}{dx}$$

H.O.T. can be ignored

$$\varepsilon(x) \approx E(x)$$

Geometric Nonlinearity cont.

- Displacement-strain relation
  - E has a higher-order term
  - $(du/dx) \ll 1 \rightarrow \varepsilon(x) \sim E(x)$.



- Domain of integration
  - Undeformed domain Ω_0
  - Deformed domain Ω_x

$$W_{int}(\mathbf{u}, \bar{\mathbf{u}}) = \int_{\Omega} \varepsilon(\bar{\mathbf{u}}) : \sigma(\mathbf{u}) d\Omega$$

Deformed domain is unknown

Material Nonlinearity

- Linear (elastic) material

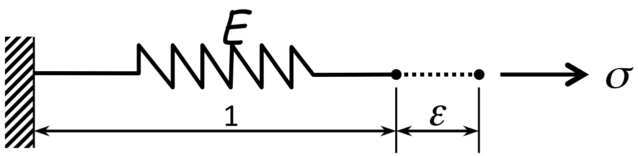
$$\{\sigma\} = [D]\{\varepsilon\}$$

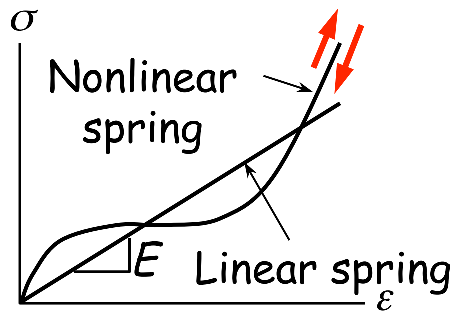
  - Only for infinitesimal deformation

- Nonlinear (elastic) material

More generally, $\{\sigma\} = \{f(\varepsilon)\}$

  - [D] is not a constant but depends on deformation
  - Stress by differentiating **strain energy density U** $\rightarrow \sigma = \dfrac{dU}{d\varepsilon}$
  - Linear material: $U = \dfrac{1}{2}E\varepsilon^2 \qquad \sigma = \dfrac{dU}{d\varepsilon} = E\varepsilon$
  - Stress is a function of strain (deformation): potential, path independent

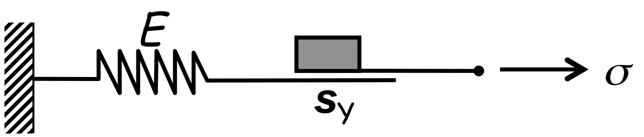


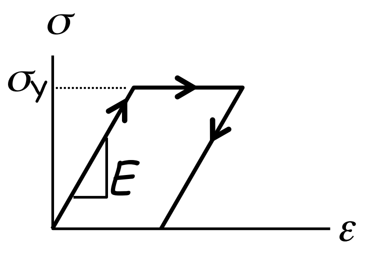


Linear and nonlinear elastic spring models

Material Nonlinearity cont.

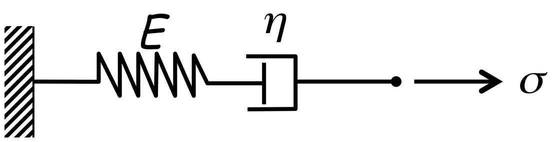
- Elasto-plastic material (energy dissipation occurs)
  - Friction plate only support stress up to σ_y
  - Stress cannot be determined from deformation alone
  - History of loading path is required: path-dependent

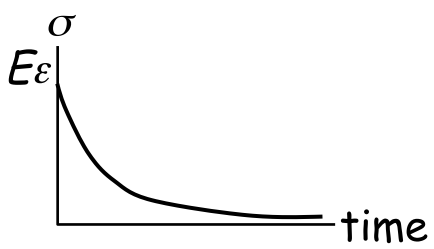




Elasto-plastic spring model

- Visco-elastic material
  - Time-dependent behavior
  - Creep, relaxation

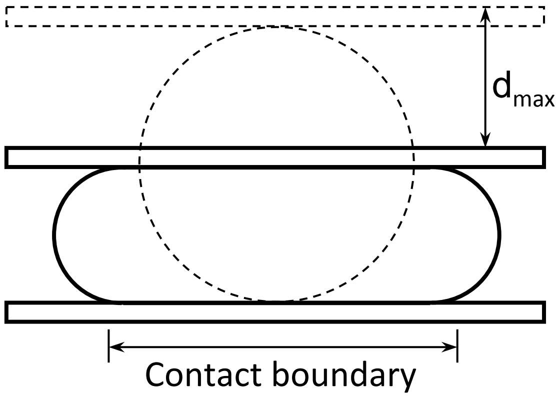


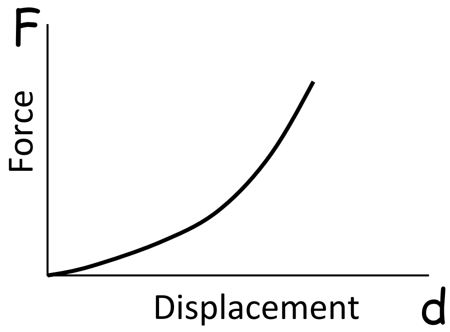


Visco-elastic spring model

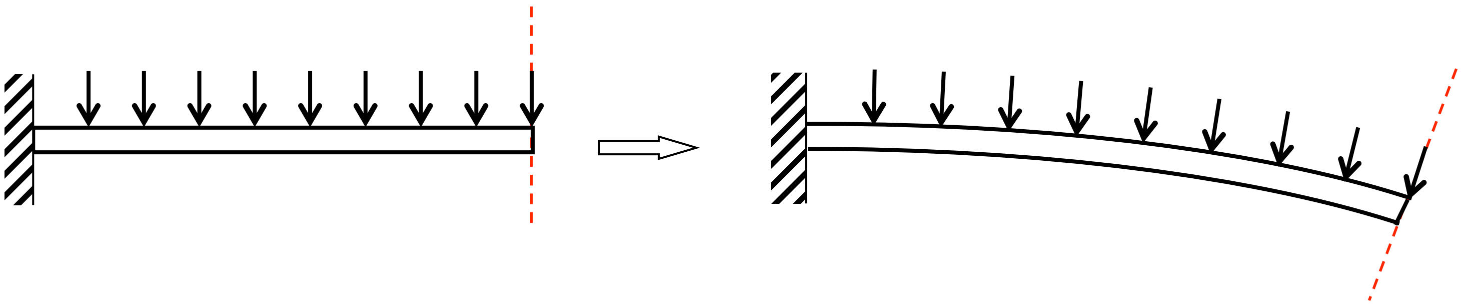
Boundary and Force Nonlinearities

- Nonlinear displacement BC (kinematic nonlinearity)
  - Contact problems, displacement dependent conditions





- Nonlinear force BC (Kinetic nonlinearity)



Mild vs. Rough Nonlinearity

- **Mild** Nonlinear Problems
  - Continuous, **history-independent** nonlinear relations between stress and strain
  - Nonlinear elasticity, Geometric nonlinearity, and deformation-dependent loads
- **Rough** Nonlinear Problems
  - Equality and/or inequality constraints in constitutive relations
  - **History-dependent** nonlinear relations between stress and strain
  - Elastoplasticity and contact problems

Nonlinear Finite Element Equations

- Equilibrium between internal and external forces

$$\boxed{\mathbf{f}_{inertia}(\ddot{\mathbf{d}}) + \mathbf{p}(\mathbf{d}) = \mathbf{f}(\mathbf{d})}$$

Inertia Internal External

Linear problems

$$[\mathbf{M}]\{\ddot{\mathbf{d}}\} + [\mathbf{K}]\{\mathbf{d}\} = \{\mathbf{f}\}$$

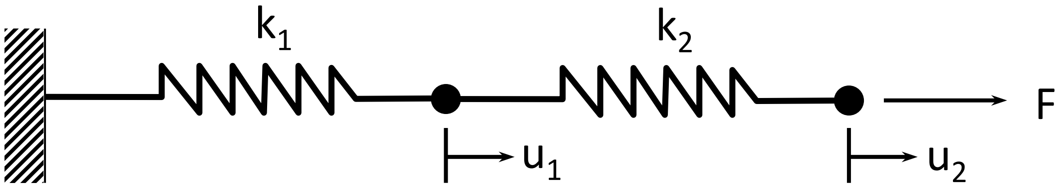
- Kinetic and kinematic nonlinearities
  - Appears on the boundary
  - Handled by displacements and forces (global, explicit)
  - Relatively easy to understand (Not easy to implement though)
- Material & geometric nonlinearities
  - Appears in the domain
  - Depends on stresses and strains (local, implicit)

Solution Procedure

We can only solve for linear problems …

Example 1 – Nonlinear Springs

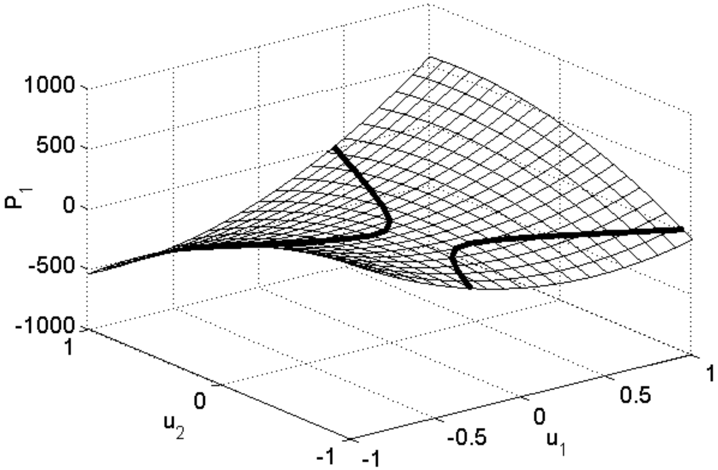
- Spring constants
  - $k_1 = 50 + 500u$
    $k_2 = 100 + 200u$

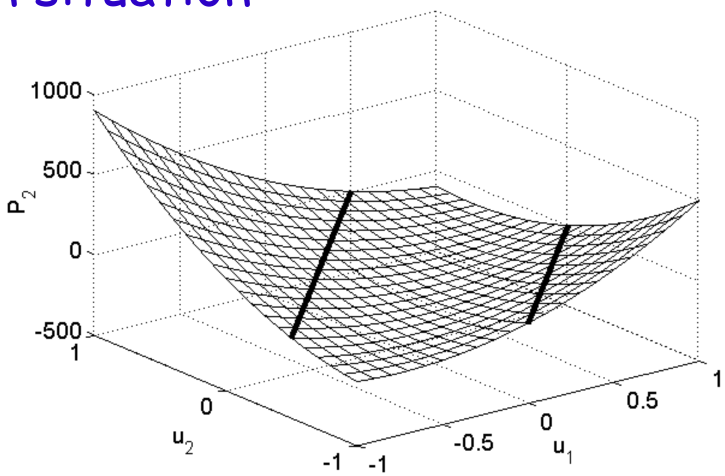


- Governing equation

$$\begin{cases} 300u_1^2 + 400u_1u_2 - 200u_2^2 + 150u_1 - 100u_2 = 0 & P_1 \\ 200u_1^2 - 400u_1u_2 + 200u_2^2 - 100u_1 + 100u_2 = 100 & P_2 \end{cases}$$

  - Solution is in the intersection between two zero contours
  - Multiple solutions may exist
  - No solution exists in a certain situation

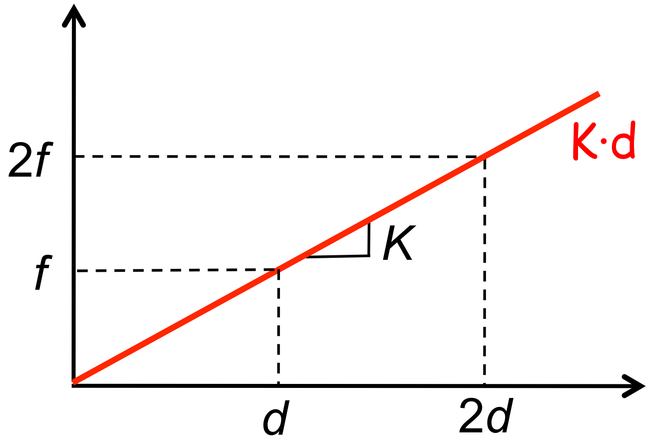




Solution Procedure

- Linear Problems

$$\mathbf{K}\cdot\mathbf{d}=\mathbf{f} \quad \text{or} \quad \mathbf{p}(\mathbf{d})=\mathbf{f}$$



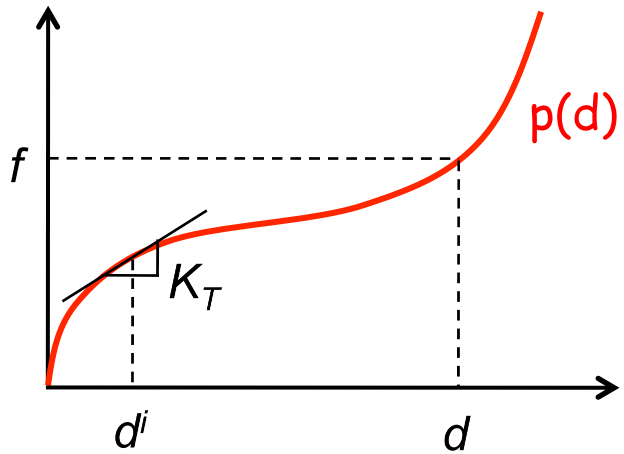
  - Stiffness matrix **K** is constant

$$\mathbf{p}(\mathbf{d}_1+\mathbf{d}_2)=\mathbf{p}(\mathbf{d}_1)+\mathbf{p}(\mathbf{d}_2)$$
$$\mathbf{p}(\alpha\mathbf{d})=\alpha\mathbf{p}(\mathbf{d})=\alpha\mathbf{f}$$

  - If the load is doubled, displacement is doubled, too
  - **Superposition** is possible

- Nonlinear Problems

$$\mathbf{p}(\mathbf{d})=\mathbf{f}, \quad \mathbf{p}(2\mathbf{d})\neq 2\mathbf{f}$$



  - How to find **d** for a given **f**?

Incremental Solution Procedure

Newton-Raphson Method

- Most popular method
- Assume $\mathbf{d}^i$ at i-th iteration is known
- Looking for $\mathbf{d}^{i+1}$ from first-order Taylor series expansion

$$\mathbf{p}(\mathbf{d}^{i+1}) \approx \mathbf{p}(\mathbf{d}^i) + \mathbf{K}_T^i(\mathbf{d}^i) \cdot \Delta\mathbf{d}^i = \mathbf{f}$$

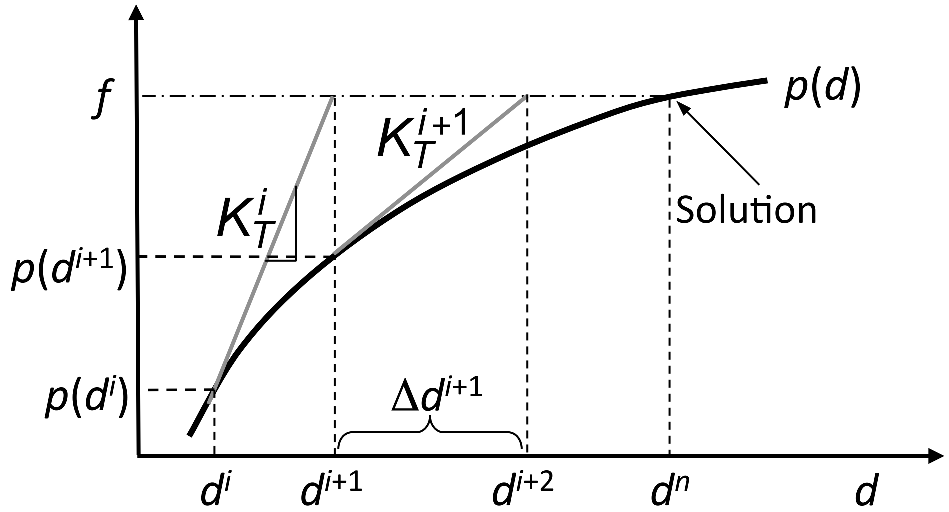
– $\mathbf{K}_T^i(\mathbf{d}^i) \equiv \left(\dfrac{\partial \mathbf{p}}{\partial \mathbf{d}}\right)^i$: **Jacobian matrix or Tangent stiffness matrix**

- **Solve for incremental solution**

$$\mathbf{K}_T^i \Delta\mathbf{d}^i = \mathbf{f} - \mathbf{p}(\mathbf{d}^i)$$

- Update solution

$$\mathbf{d}^{i+1} = \mathbf{d}^i + \Delta\mathbf{d}^i$$



N-R Method cont.

- Observations:
  - **Second-order convergence** near the solution (Fastest method!) $\lim_{n\to\infty} \frac{|u_{exact} - u_{n+1}|}{|u_{exact} - u_n|^2} = c$
  - Tangent stiffness $\mathbf{K}_T^i(\mathbf{d}^i)$ is not constant
  - The matrix equation solves for incremental displacement $\Delta\mathbf{d}^i$
  - RHS is not a force but a **residual force** $\mathbf{r}^i \equiv \mathbf{f} - \mathbf{p}(\mathbf{d}^i)$
  - Iteration stops when conv < tolerance

$$\text{conv} = \frac{\|\mathbf{r}^{i+1}\|_2}{1 + \|\mathbf{f}\|_2} \qquad \text{Or,} \qquad \text{conv} = \frac{\|\Delta\mathbf{d}^{i+1}\|_2}{1 + \|\Delta\mathbf{d}^0\|_2}$$

N-R Algorithm

1. Set tolerance = 0.001, k = 0, max_iter = 20, and initial estimate $d^k = d_0$
2. Calculate residual $r^k = f - p(d^k)$
3. Calculate conv. If conv < tolerance, stop
4. If k > max_iter, stop with error message
5. Calculate Jacobian matrix K_T^k at u^k
6. If the determinant of K_T^k is zero, stop with error messg.
7. Calculate solution increment Δd^k
8. Update solution by $d^{k+1} = d^k + \Delta d^k$
9. Set $d^k = d^{k+1}$
10. Go to Step 2

Example 2 - N-R Method

$$\mathbf{p}(\mathbf{d}) \equiv \begin{Bmatrix} d_1 + d_2 \\ d_1^2 + d_2^2 \end{Bmatrix} = \begin{Bmatrix} 3 \\ 9 \end{Bmatrix} \equiv \mathbf{f} \qquad \mathbf{d}^0 = \begin{Bmatrix} 1 \\ 5 \end{Bmatrix} \qquad \mathbf{p}(\mathbf{d}^0) = \begin{Bmatrix} 6 \\ 26 \end{Bmatrix}$$

$$\mathbf{K}_T = \frac{\partial \mathbf{p}}{\partial \mathbf{d}} = \begin{bmatrix} 1 & 1 \\ 2d_1 & 2d_2 \end{bmatrix} \qquad \mathbf{r}^0 = \mathbf{f} - \mathbf{p}(\mathbf{d}^0) = \begin{Bmatrix} -3 \\ -17 \end{Bmatrix}$$

- Iteration 1

$$\begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix} \begin{Bmatrix} \Delta d_1^0 \\ \Delta d_2^0 \end{Bmatrix} = \begin{Bmatrix} -3 \\ -17 \end{Bmatrix} \quad \Longrightarrow \quad \begin{Bmatrix} \Delta d_1^0 \\ \Delta d_2^0 \end{Bmatrix} = \begin{Bmatrix} -1.625 \\ -1.375 \end{Bmatrix}$$

$$\mathbf{d}^1 = \mathbf{d}^0 + \Delta \mathbf{d}^0 = \begin{Bmatrix} -0.625 \\ 3.625 \end{Bmatrix}$$

$$\mathbf{r}^1 = \mathbf{f} - \mathbf{P}(\mathbf{d}^1) = \begin{Bmatrix} 0 \\ -4.531 \end{Bmatrix}$$

Example 2 – N-R Method cont.

- Iteration 2

$$\begin{bmatrix} 1 & 1 \\ -1.25 & 7.25 \end{bmatrix} \begin{Bmatrix} \Delta d_1^1 \\ \Delta d_2^1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -4.531 \end{Bmatrix} \Longrightarrow \begin{Bmatrix} \Delta d_1^1 \\ \Delta d_2^1 \end{Bmatrix} = \begin{Bmatrix} 0.533 \\ -0.533 \end{Bmatrix}$$

$$\mathbf{d}^2 = \mathbf{d}^1 + \Delta\mathbf{d}^1 = \begin{Bmatrix} -0.092 \\ 3.092 \end{Bmatrix} \qquad \mathbf{r}^2 = \mathbf{f} - \mathbf{p}(\mathbf{d}^2) = \begin{Bmatrix} 0 \\ -0.568 \end{Bmatrix}$$

- Iteration 3

$$\begin{bmatrix} 1 & 1 \\ -0.184 & 6.184 \end{bmatrix} \begin{Bmatrix} \Delta d_1^2 \\ \Delta d_2^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -0.568 \end{Bmatrix} \Longrightarrow \begin{Bmatrix} \Delta d_1^2 \\ \Delta d_2^2 \end{Bmatrix} = \begin{Bmatrix} 0.089 \\ -0.089 \end{Bmatrix}$$

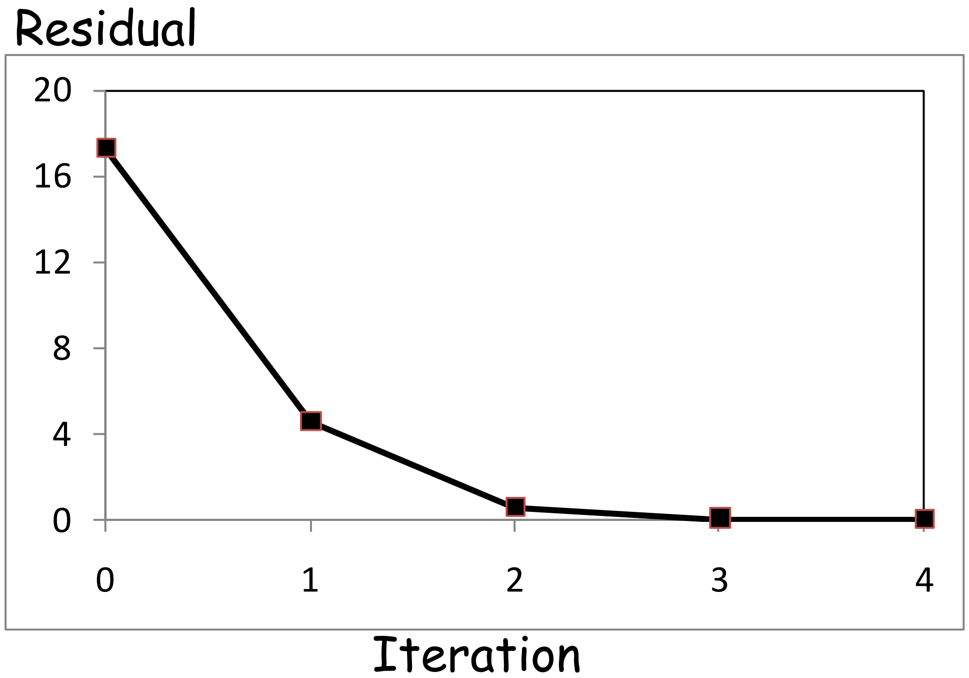
$$\mathbf{d}^3 = \mathbf{d}^2 + \Delta\mathbf{d}^2 = \begin{Bmatrix} -0.003 \\ 3.003 \end{Bmatrix} \qquad \mathbf{r}^3 = \mathbf{f} - \mathbf{p}(\mathbf{d}^3) = \begin{Bmatrix} 0 \\ -0.016 \end{Bmatrix}$$

Example 2 - N-R Method cont.

- Iteration 4

$$\begin{bmatrix} 1 & 1 \\ -0.005 & 6.005 \end{bmatrix} \begin{Bmatrix} \Delta d_1^3 \\ \Delta d_2^3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -0.016 \end{Bmatrix} \Rightarrow \begin{Bmatrix} \Delta d_1^3 \\ \Delta d_2^3 \end{Bmatrix} = \begin{Bmatrix} 0.003 \\ -0.003 \end{Bmatrix}$$

$$\mathbf{d}^4 = \mathbf{d}^3 + \Delta\mathbf{d}^3 = \begin{Bmatrix} -0.000 \\ 3.000 \end{Bmatrix} \qquad \mathbf{r}^4 = \mathbf{f} - \mathbf{p}(\mathbf{d}^4) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

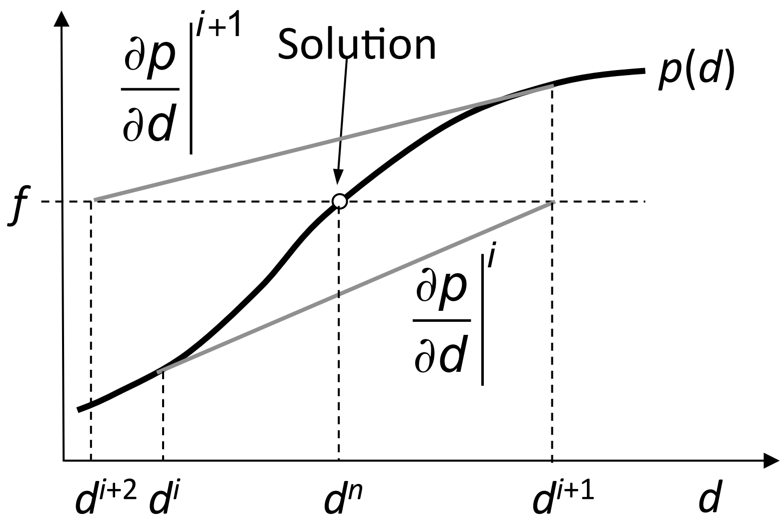


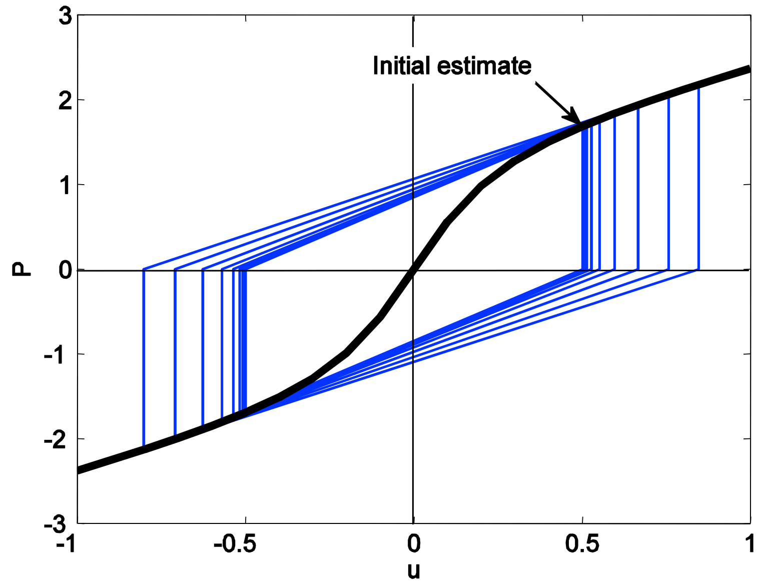
Iteration k	$\|\|r^k\|\|$
0	17.263
1	4.531
2	0.016
3	0.0

Quadratic convergence

When N-R Method Does Not Converge?

- Difficulties
  - **Convergence is not always guaranteed**
  - Automatic load step control and/or line search techniques are often used
  - Difficult/expensive to calculate $\mathbf{K}_T^i(\mathbf{d}^i)$



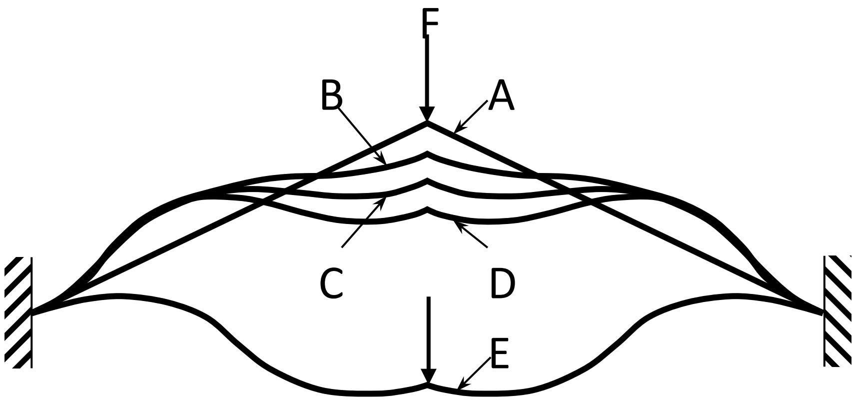


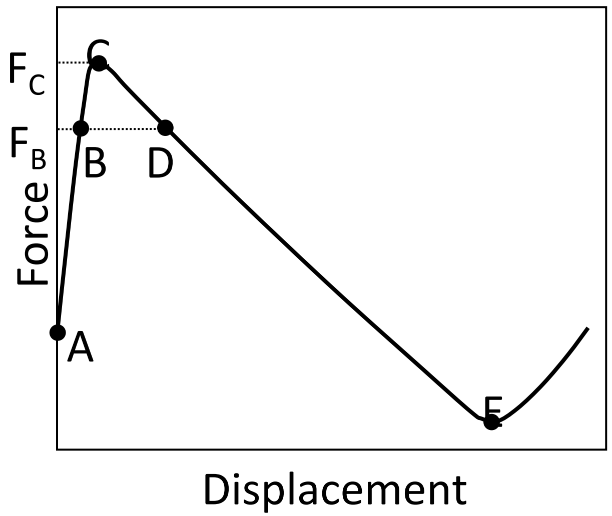
When N-R Method Does Not Converge? cont.

- Convergence difficulty occurs when
  - Jacobian matrix is not positive-definite

    P.D. Jacobian: in order to increase displ., force must be increased

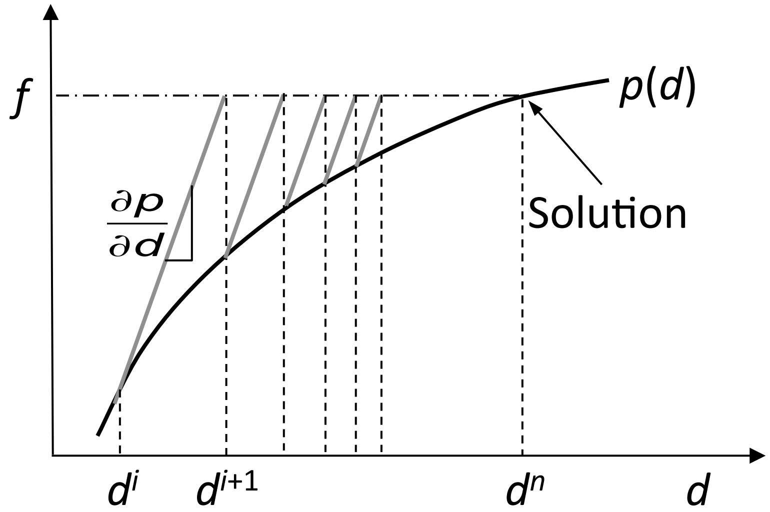
  - Bifurcation & snap-through require a special algorithm
  - Cracking of not reinforced concrete





Modified N-R Method

- Constructing $\mathbf{K}_T^i(\mathbf{d}^i)$ and solving $\mathbf{K}_T^i \Delta\mathbf{d}^i = \mathbf{r}^i$ is expensive
- Computational Costs (Let the matrix size be N x N)
  - L-U factorization ~ N^3
  - Forward/backward substitution ~ N
- **Use L-U factorized $\mathbf{K}_T^i(\mathbf{d}^i)$ repeatedly**
- More iteration is required, but each iteration is fast
- **More stable than N-R method**
- Hybrid N-R method



Example 3 – Modified N-R Method

- Solve the same problem using modified N-R method

$$\mathbf{p}(\mathbf{d}) \equiv \begin{Bmatrix} d_1 + d_2 \\ d_1^2 + d_2^2 \end{Bmatrix} = \begin{Bmatrix} 3 \\ 9 \end{Bmatrix} \equiv \mathbf{f} \qquad \mathbf{d}^0 = \begin{Bmatrix} 1 \\ 5 \end{Bmatrix} \qquad \mathbf{P}(\mathbf{d}^0) = \begin{Bmatrix} 6 \\ 26 \end{Bmatrix}$$

$$\mathbf{K}_T = \frac{\partial \mathbf{p}}{\partial \mathbf{d}} = \begin{bmatrix} 1 & 1 \\ 2d_1 & 2d_2 \end{bmatrix} \qquad \mathbf{r}^0 = \mathbf{f} - \mathbf{p}(\mathbf{d}^0) = \begin{Bmatrix} -3 \\ -17 \end{Bmatrix}$$

- Iteration 1

$$\begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix} \begin{Bmatrix} \Delta d_1^0 \\ \Delta d_2^0 \end{Bmatrix} = \begin{Bmatrix} -3 \\ -17 \end{Bmatrix} \Longrightarrow \begin{Bmatrix} \Delta d_1^0 \\ \Delta d_2^0 \end{Bmatrix} = \begin{Bmatrix} -1.625 \\ -1.375 \end{Bmatrix}$$

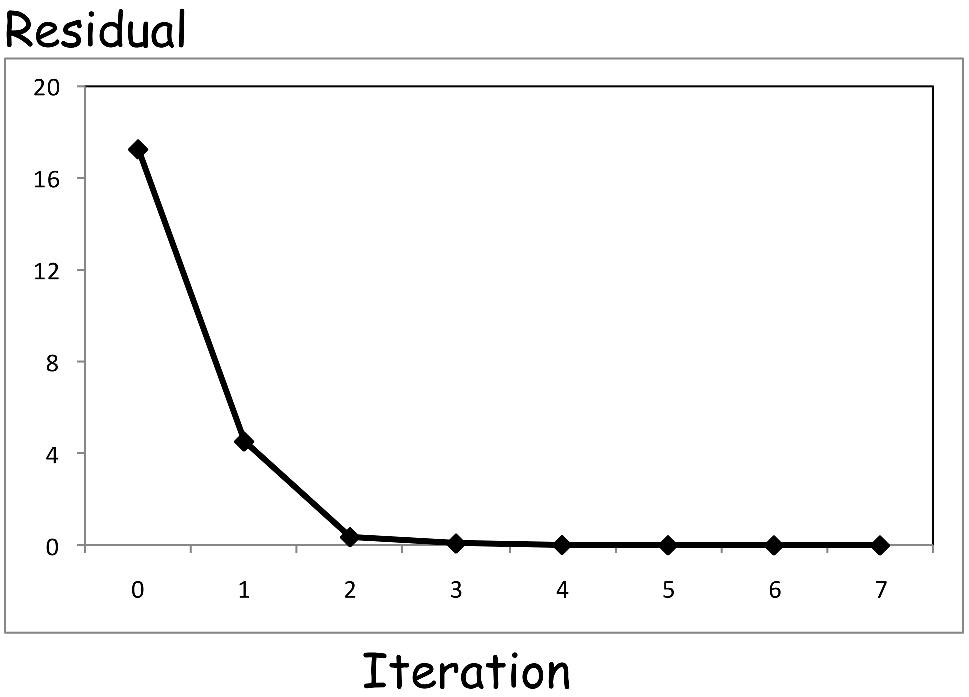
$$\mathbf{d}^1 = \mathbf{d}^0 + \Delta\mathbf{d}^0 = \begin{Bmatrix} -0.625 \\ 3.625 \end{Bmatrix} \qquad \mathbf{r}^1 = \mathbf{f} - \mathbf{p}(\mathbf{d}^1) = \begin{Bmatrix} 0 \\ -4.531 \end{Bmatrix}$$

Example 3 - Modified N-R Method cont.

- Iteration 2

$$\begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix} \begin{Bmatrix} \Delta d_1^1 \\ \Delta d_2^1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -4.531 \end{Bmatrix} \Rightarrow \begin{Bmatrix} \Delta d_1^1 \\ \Delta d_2^1 \end{Bmatrix} = \begin{Bmatrix} 0.566 \\ -0.566 \end{Bmatrix}$$

$$\mathbf{d}^2 = \mathbf{d}^1 + \Delta\mathbf{d}^1 = \begin{Bmatrix} -0.059 \\ 3.059 \end{Bmatrix} \qquad \mathbf{r}^2 = \mathbf{f} - \mathbf{p}(\mathbf{d}^2) = \begin{Bmatrix} 0 \\ -0.358 \end{Bmatrix}$$

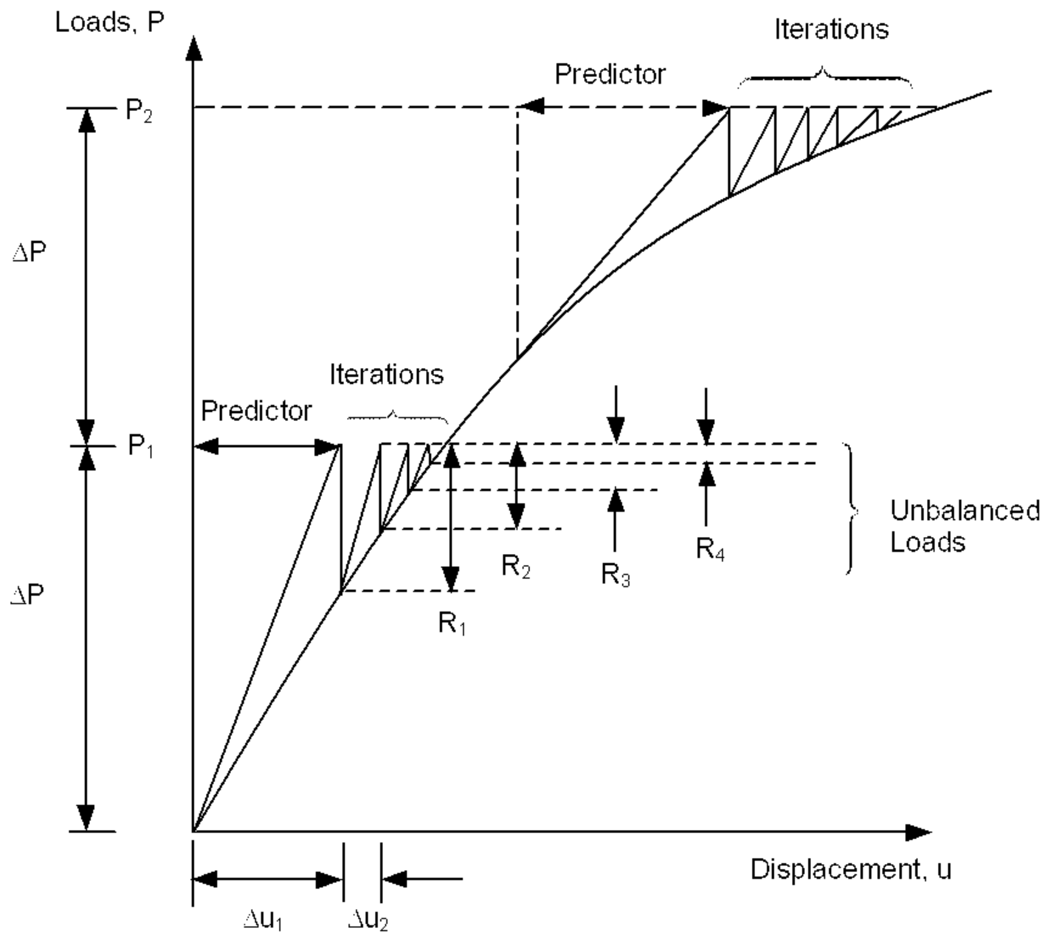


Iteration k	$\|\|r^k\|\|$
0	17.263
1	4.5310
2	0.3584
3	0.0831
4	0.0204
5	0.0051
6	0.0013
7	0.0003

NR implementation in Solid Mechanics

Incremental Force Method

- N-R method converges fast if the initial estimate is close to the solution
- Solid mechanics: initial estimate = undeformed shape
- Convergence difficulty occurs when the applied load is large (deformation is large)
- IFM: apply loads in increments. Use the solution from the previous increment as an initial estimate
- Commercial programs call it "**Load Increment**" or "**Time Increment**"



Incremental Force Method cont.

- Load increment does not have to be uniform
  - Critical part has smaller increment size
- Solutions in the intermediate load increments
  - History of the response can provide insight into the problem
  - Estimating the bifurcation point or the critical load
  - Load increments greatly affect the accuracy in path-dependent problems

Load Increment implementation

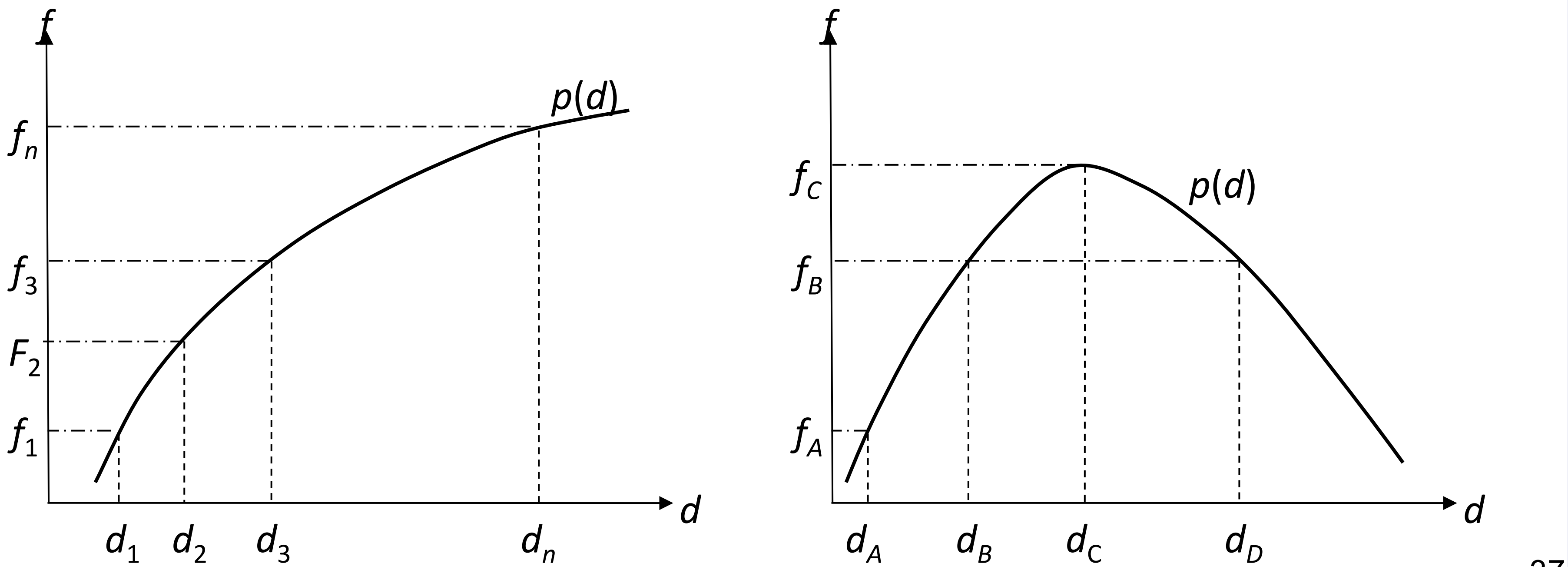
- Use "Time" to represent load level
  - In a static problem, "Time" means a pseudo-time
  - Required Starting time, (T_{start}), Ending time (T_{end}) and increment
  - Load is gradually increased from zero at T_{start} and full load at T_{end}
  - Load magnitude at load increment T^n:

$$f^n = \frac{T^n - T_{start}}{T_{end} - T_{start}} f \qquad\qquad T^n = n \times \Delta T \leq T_{end}$$

- Automatic time stepping
  - Increase/decrease next load increment based on the number of convergence iteration at the current load
  - User provide initial load increment, minimum increment, and maximum increment
  - Bisection of load increment when not converged

Force Control vs. Displacement Control

- Force control: gradually increase applied forces and find equilibrium configuration
- Displ. control: gradually increase prescribed displacements
  - Applied load can be calculated as a reaction
  - More stable than force control.
  - Useful for softening, contact, snap-through, etc.



Nonlinear Solution Steps

1. Initialization: $\mathbf{d}^0 = \mathbf{0};\ i = 0$

2. Residual Calculation $\mathbf{r}^i = \mathbf{f} - \mathbf{p}(\mathbf{d}^i)$

3. Convergence Check (If converged, stop)

4. Linearization
   - Calculate tangent stiffness $\mathbf{K}_T^i(\mathbf{d}^i)$

5. Incremental Solution:
   - Solve $\mathbf{K}_T^i(\mathbf{d}^i)\Delta\mathbf{d}^i = \mathbf{r}^i$

6. State Determination
   - Update displacement and stress $\mathbf{d}^{i+1} = \mathbf{d}^i + \Delta\mathbf{d}^i$, $\sigma^{i+1} = \sigma^i + \Delta\sigma^i$

7. Go To Step 2

Nonlinear Solution Steps cont.

- State determination
  - For a given displ $\mathbf{d}^k$, determine current state (strain, stress, etc)

$$\mathbf{u}^k(\mathbf{x}) = \mathbf{N}(\mathbf{x}) \cdot \mathbf{d}^k \qquad \varepsilon^k = \mathbf{B} \cdot \mathbf{d}^k \qquad \sigma^k = f(\varepsilon^k)$$

  - Sometimes, stress cannot be determined using strain alone

- Residual calculation (static case)
  - Applied nodal force – Nodal forces due to internal stresses

Weak form: $$\int_\Omega \varepsilon(\bar{\mathbf{u}})^T \sigma \, d\Omega = \int_{\Gamma_s} \bar{\mathbf{u}}^T \mathbf{t} \, d\Gamma + \int_\Omega \bar{\mathbf{u}}^T \mathbf{f}^b \, d\Omega, \quad \forall \bar{\mathbf{u}} \in \mathbb{Z}$$

Discretization: $$\bar{\mathbf{d}}^T \left(\int_\Omega \mathbf{B}^T \sigma \, d\Omega = \int_{\Gamma_s} \mathbf{N}^T \mathbf{t} \, d\Gamma + \int_\Omega \mathbf{N}^T \mathbf{f}^b \, d\Omega \right), \quad \forall \bar{\mathbf{d}} \in \mathbb{Z}_h$$

Residual: $$\mathbf{r}^k = \underbrace{\int_{\Gamma_s} \mathbf{N}^T \mathbf{t} \, d\Gamma + \int_\Omega \mathbf{N}^T \mathbf{f}^b \, d\Omega}_{\mathbf{f}} - \underbrace{\int_\Omega \mathbf{B}^T \sigma^k \, d\Omega}_{\mathbf{p}(\mathbf{d})}$$

Particularization to Linear Elastic Material

- Governing equation (Scalar equation)

$$\int_\Omega \varepsilon(\bar{\mathbf{u}})^T \sigma \, d\Omega = \int_{\Gamma_s} \bar{\mathbf{u}}^T \mathbf{t} \, d\Gamma + \int_\Omega \bar{\mathbf{u}}^T \mathbf{f}^b \, d\Omega$$

$$\bar{\mathbf{u}} = \mathbf{N} \cdot \bar{\mathbf{d}}$$

$$\varepsilon(\bar{\mathbf{u}}) = \mathbf{B} \cdot \bar{\mathbf{d}}$$

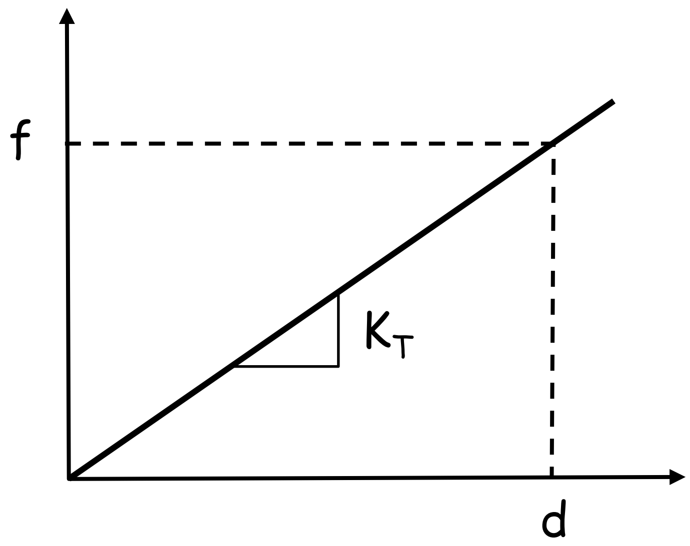
- Collect $\bar{\mathbf{d}}$

$$\bar{\mathbf{d}}^T \left(\underbrace{\int_\Omega \mathbf{B}^T \sigma \, d\Omega}_{\mathbf{p}(\mathbf{d})} = \underbrace{\int_{\Gamma_s} \mathbf{N}^T \mathbf{t} \, d\Gamma + \int_\Omega \mathbf{N}^T \mathbf{f}^b \, d\Omega}_{\mathbf{f}} \right)$$

- Residual $\quad \mathbf{r} = \mathbf{f} - \mathbf{p}(\mathbf{d})$
- Linear elastic material

$$\sigma = \mathbf{D} \cdot \varepsilon = \mathbf{D} \cdot \mathbf{B} \cdot \mathbf{d}$$

$$\mathbf{K}_T = \frac{\partial \mathbf{p}(\mathbf{d})}{\partial \mathbf{d}} = \int_\Omega \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega$$



Example 4 – Nonlinear Bar

- Rubber bar $\sigma = E\tan^{-1}(m\varepsilon)$

$$\bar{\mathbf{d}} = \begin{Bmatrix} \bar{d}_1 \\ \bar{d}_2 \end{Bmatrix}$$

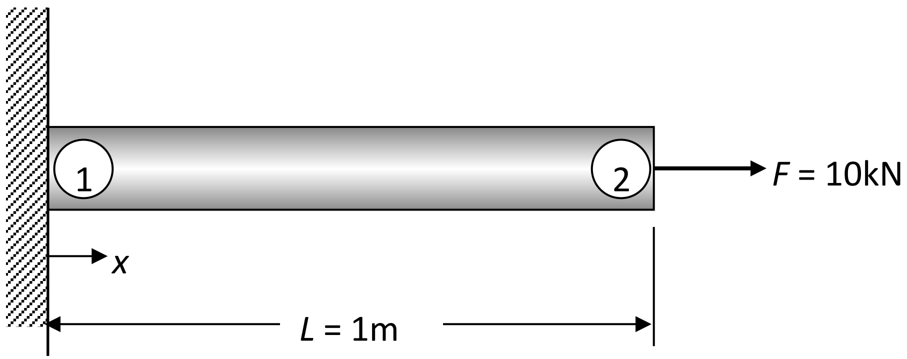
- Discrete weak form $\bar{\mathbf{d}}^T \int_0^L \mathbf{B}^T \sigma A\,dx = \bar{\mathbf{d}}^T \mathbf{F}$

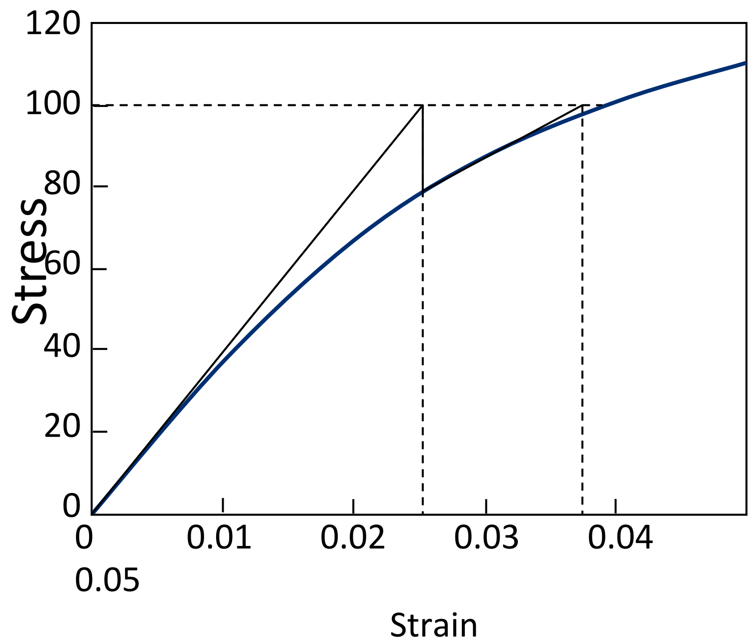
$$\mathbf{F} = \begin{Bmatrix} R \\ F \end{Bmatrix}$$

- Scalar equation $r = F - \int_0^L \frac{\sigma A}{L} dx$

$$\Rightarrow r = F - \sigma(d)A$$

$$\mathbf{B} = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix}$$





Example 4 – Nonlinear Bar cont.

- Jacobian

$$\frac{dp}{dd} = \frac{d\sigma(d)}{dd}A = \frac{d\sigma}{d\varepsilon}\frac{d\varepsilon}{dd}A = \frac{1}{L}mAE\cos^2\left(\frac{\sigma}{E}\right)$$

- N-R equation

$$\left[\frac{1}{L}mAE\cos^2\left(\frac{\sigma^k}{E}\right)\right]\Delta d^k = F - \sigma^k A$$

- Iteration 1

$$\frac{mAE}{L}\Delta d^0 = F$$

$$d^1 = d^0 + \Delta d^0 = 0.025\text{m}$$

$$\varepsilon^1 = d^1 / L = 0.025$$

$$\sigma^1 = E\tan^{-1}(m\varepsilon^1) = 78.5\text{MPa}$$

- Iteration 2

$$\left[\frac{mAE}{L}\cos^2\left(\frac{\sigma^1}{E}\right)\right]\Delta d^1 = F - \sigma^1 A$$

$$d^2 = d^1 + \Delta d^1 = 0.0357\text{m}$$

$$\varepsilon^2 = d^2 / L = 0.0357$$

$$\sigma^2 = E\tan^{-1}(m\varepsilon^2) = 96\text{MPa}$$

N-R or Modified N-R?

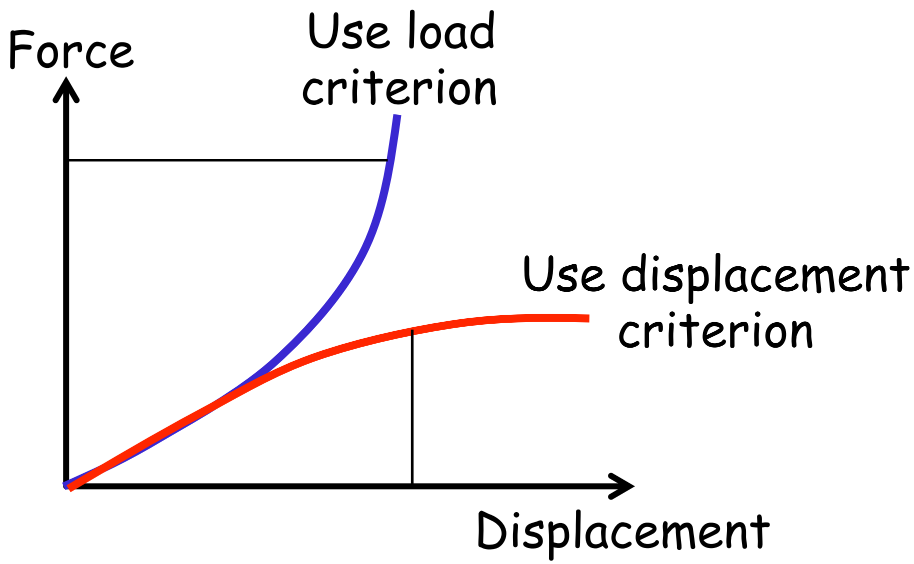
- It is always recommended to use the Incremental Force Method
  - Mild nonlinear: ~10 increments
  - Rough nonlinear: 20 ~ 100 increments
  - For rough nonlinear problems, analysis results depends on increment size
- Within an increment, N-R or modified N-R can be used
  - N-R method calculates K_T at every iteration
  - Modified N-R method calculates K_T once at every increment
  - N-R is better when: mild nonlinear problem, tight convergence criterion
  - Modified N-R is better when: computation is expensive, small increment size, and when N-R does not converge well
- Many FE programs provide automatic stiffness update option
  - Depending on convergence criteria used, material status change, etc

Accuracy vs. Convergence

- Nonlinear solution procedure requires
  - Internal force $\mathbf{p}(\mathbf{d})$
  - Tangent stiffness $\mathbf{K}_T(\mathbf{d}) = \dfrac{\partial \mathbf{p}(\mathbf{d})}{\partial \mathbf{d}}$
  - They are often implemented in the same routine
- Internal force $\mathbf{p}(\mathbf{d})$ needs to be accurate
  - We solve equilibrium of $\mathbf{p}(\mathbf{d}) = \mathbf{f}$
- Tangent stiffness $\mathbf{K}_T(\mathbf{d})$ contributes to convergence
  - Accurate $\mathbf{K}_T(\mathbf{d})$ provides quadratic convergence near the solution
  - Approximate $\mathbf{K}_T(\mathbf{d})$ requires more iteration to converge
  - Wrong $\mathbf{K}_T(\mathbf{d})$ causes lack of convergence

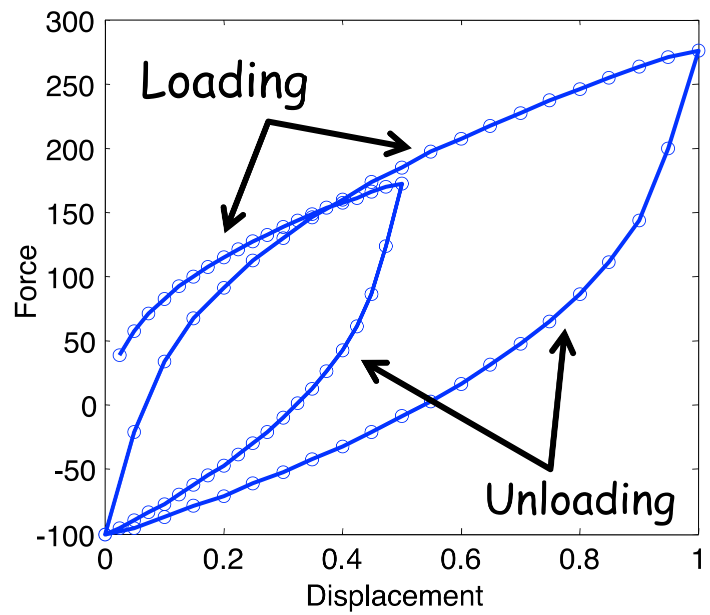
Convergence Criteria

- Most analysis programs provide three convergence criteria
  - Work, displacement, load (residual)
  - Work = displacement x load
  - At least two criteria needs to be converged
- Traditional convergence criterion is load (residual)
  - Equilibrium between internal and external forces **p**(**d**) = **f**(**d**)
- Use displacement criterion for load insensitive system

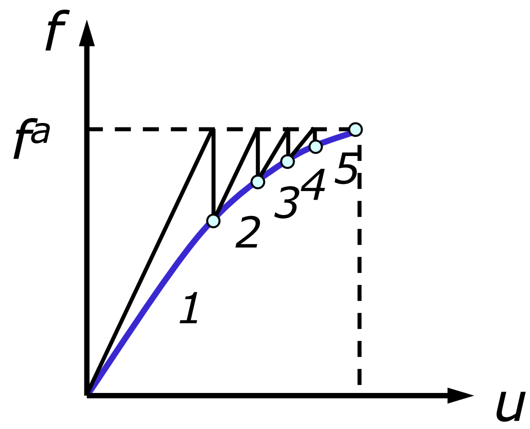


Solution Strategies

- Load Increment (substeps)
  - Linear analysis concerns max load
  - Nonlinear analysis depends on load path (history)
  - Applied load is gradually increased within a load step
  - Follow load path, improve accuracy, and easy to converge



- Convergence Iteration
  - Within a load increment, an iterative method (e.g., NR method) is used to find nonlinear solution
  - Bisection, linear search, stabilization, etc

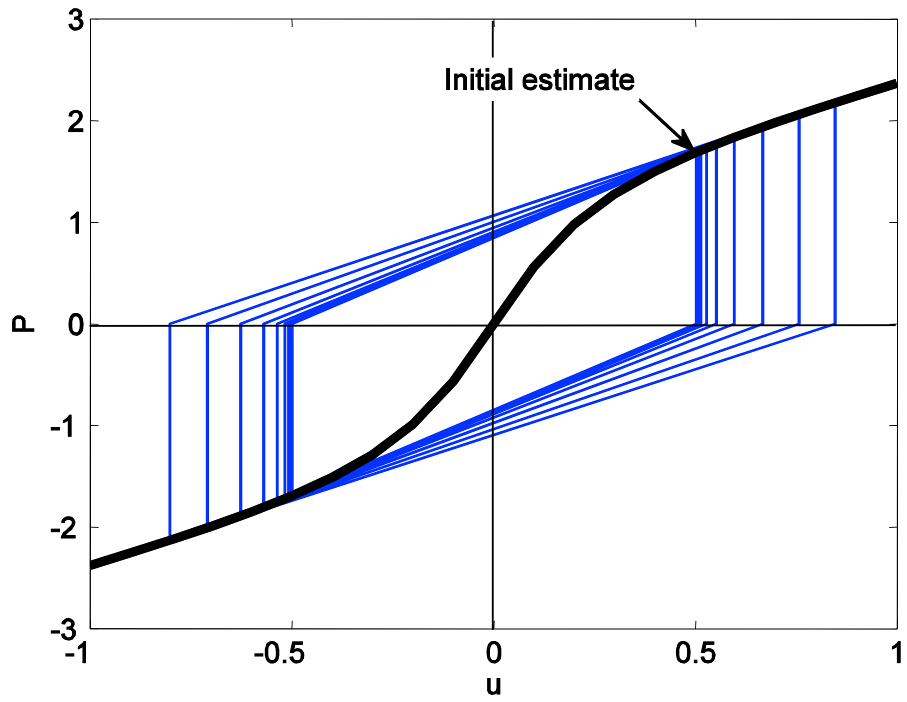


Solution Strategies cont.

- Automatic (Variable) Load Increment
  - Also called Automatic Time Stepping
  - Load increment may not be uniform
  - When convergence iteration diverges, the load increment is halved
  - If a solution converges in less than 4 iterations, increase time increment by 25%
  - If a solution converges in more than 8 iterations, decrease time increment by 25%
- Subincrement (or bisection)
  - When iterations do not converge at a given increment, analysis goes back to previously converged increment and the load increment is reduced by half
  - This process is repeated until max number of subincrements

When nonlinear analysis does not converge

- NR method assumes a constant curvature locally
- When a sign of curvature changes around the solution, NR method oscillates or diverges
- Often the residual changes sign between iterations
- Line search can help to converge



$$p(u) = u + \tan^{-1}(5u)$$

$$\frac{dp}{du} = 1 + 5\cos^2(\tan^{-1}(5u))$$

When nonlinear analysis does not converge

- Displacement-controlled vs. force-controlled procedure
  - Almost all linear problems are force-controlled
  - Displacement-controlled procedure is more stable for nonlinear analysis
  - Use reaction forces to calculate applied forces