A short course on

Nonlinear Finite Element Analysis

This video:

Newton-Raphson and Load Incrementation

Terje's Toolbox is freely available at terje.civil.ubc.ca
It is created and maintained by Professor Terje Haukaas, Ph.D., P.Eng.,
Department of Civil Engineering, The University of British Columbia (UBC), Vancouver, Canada

Notation from Linear Analysis

Equilibrium equations: $\mathbf{Ku} = \mathbf{F}$

Index notation: $K_{ij}\, u_j = F_i$

K_{ij} = force along DOF i due to a unit disp./rot. along DOF j

u_j = unknown displacement or rotation along DOF j

F_i = force along DOF i due to external loads

Split member forces and point loads: $\mathbf{Ku} + \bar{\mathbf{F}} = \grave{\mathbf{F}}$

Total load vector: $\mathbf{Ku} = \grave{\mathbf{F}} - \bar{\mathbf{F}} = \mathbf{F}$

Member end forces after solving equilibrium equations: $\mathbf{F} = \mathbf{Ku} + \bar{\mathbf{F}}$

Nonlinear Analysis

Equilibrium equations: $\tilde{\mathbf{F}}(\mathbf{u}) = \mathbf{F}$

Internal resisting forces (nonlinear function of $\mathbf{u}$)

Externally applied loads

Configurations

$$\tilde{\mathbf{F}}_f \quad — \quad \mathbf{K}_f = \mathbf{T}_{af}^{\mathrm{T}}\mathbf{K}_a\mathbf{T}_{af} \quad — \quad \mathbf{u}_f$$

$$\mathbf{F}_f = \mathbf{T}_{af}^{\mathrm{T}}\mathbf{F}_a \qquad \mathbf{u}_a = \mathbf{T}_{af}\mathbf{u}_f$$

$$\tilde{\mathbf{F}}_a \quad — \quad \mathbf{K}_a = \Sigma\left(\mathbf{T}_{ga}^{\mathrm{T}}\mathbf{K}_g\mathbf{T}_{ga}\right) \quad — \quad \mathbf{u}_a$$

$$\mathbf{F}_a = \Sigma\left(\mathbf{T}_{ga}^{\mathrm{T}}\mathbf{F}_g\right) \qquad \mathbf{u}_g = \mathbf{T}_{ga}\mathbf{u}_a$$

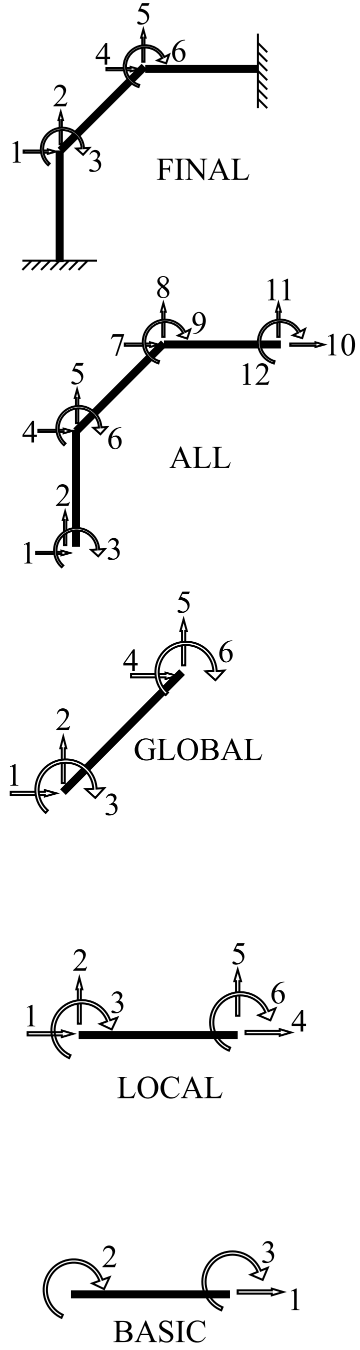
$$\tilde{\mathbf{F}}_g \quad — \quad \mathbf{K}_g = \mathbf{T}_{lg}^{\mathrm{T}}\mathbf{K}_l\mathbf{T}_{lg} \quad — \quad \mathbf{u}_g$$

$$\mathbf{F}_g = \mathbf{T}_{lg}^{\mathrm{T}}\mathbf{F}_l \qquad \mathbf{u}_l = \mathbf{T}_{lg}\mathbf{u}_g$$

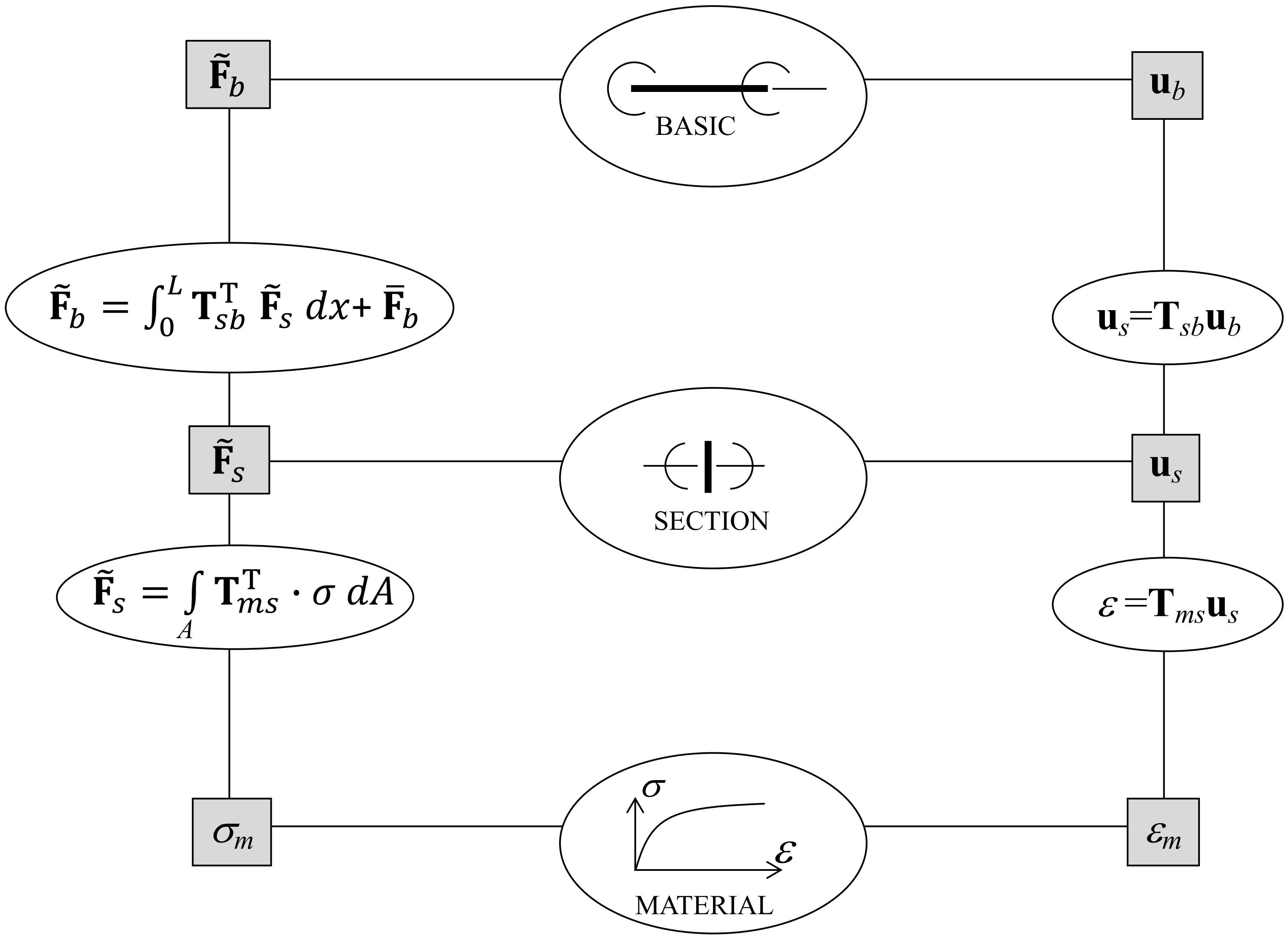
$$\tilde{\mathbf{F}}_l \quad — \quad \mathbf{K}_l = \mathbf{T}_{bl}^{\mathrm{T}}\mathbf{K}_b\mathbf{T}_{bl} \quad — \quad \mathbf{u}_l$$

$$\mathbf{F}_l = \mathbf{T}_{bl}^{\mathrm{T}}\mathbf{F}_b \qquad \mathbf{u}_b = \mathbf{T}_{bl}\mathbf{u}_l$$

$$\tilde{\mathbf{F}}_b \quad — \quad \mathbf{K}_b \quad — \quad \mathbf{u}_b$$



Added in Nonlinear Analysis



Analysis Procedure

1. Determine trial displacements, $\mathbf{u}_f$ (will soon see how)

2. Determine corresponding strain, $\varepsilon = \mathbf{T}_{ms}\mathbf{T}_{sb}\mathbf{T}_{bl}\mathbf{T}_{lg}\mathbf{T}_{ga}\mathbf{T}_{af}\mathbf{u}_f$ (can do better than $\mathbf{T}_{ga}\mathbf{T}_{af}$)

3. Determine stress for given strain from the material law, often history-dependent, i.e., hysteretic

4. Determine resisting forces: $\tilde{\mathbf{F}}_f(\mathbf{u}_f) = \mathbf{T}_{af}^T \sum \left(\mathbf{T}_{ga}^T \mathbf{T}_{lg}^T \mathbf{T}_{bl}^T \left(\int_0^L \mathbf{T}_{sb}^{\mathrm{T}} \left(\int \mathbf{T}_{ms}^{\mathrm{T}} \cdot \sigma \, dA \right) dx + \bar{\mathbf{F}}_b \right) \right)$

5. Check convergence: $\tilde{\mathbf{F}} = \mathbf{F}$?

Newton-Raphson

Equilibrium on residual form: $\tilde{\mathbf{F}}(\mathbf{u}) - \mathbf{F} = \mathbf{R} = \mathbf{0}$

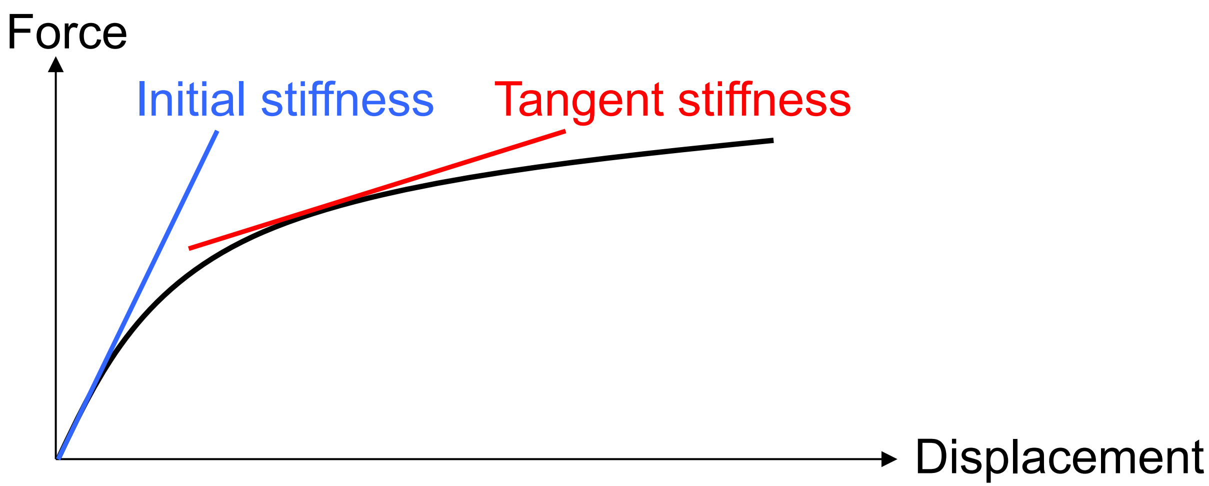
First-order Taylor expansion of the residual: $\mathbf{R}(\mathbf{u}) = \mathbf{R}(\mathbf{u}_i) + \frac{\partial \mathbf{R}(\mathbf{u}_i)}{\partial \mathbf{u}}(\mathbf{u} - \mathbf{u}_i)$

Set the residual to zero and recognize linear system of equations: $\frac{\partial \mathbf{R}(\mathbf{u}_i)}{\partial \mathbf{u}}\Delta\mathbf{u} = -\mathbf{R}(\mathbf{u}_i)$

Because **R** is a nonlinear function of $\mathbf{u}$, we iterate: $\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta\mathbf{u}$ (the trial displacements)

About the derivative: $\frac{\partial \mathbf{R}(\mathbf{u}_i)}{\partial \mathbf{u}} = \frac{\partial \tilde{\mathbf{F}}(\mathbf{u}_i)}{\partial \mathbf{u}} = \mathbf{K}_{\text{tangent}}$

Tangent Stiffness



Modified Newton-Raphson

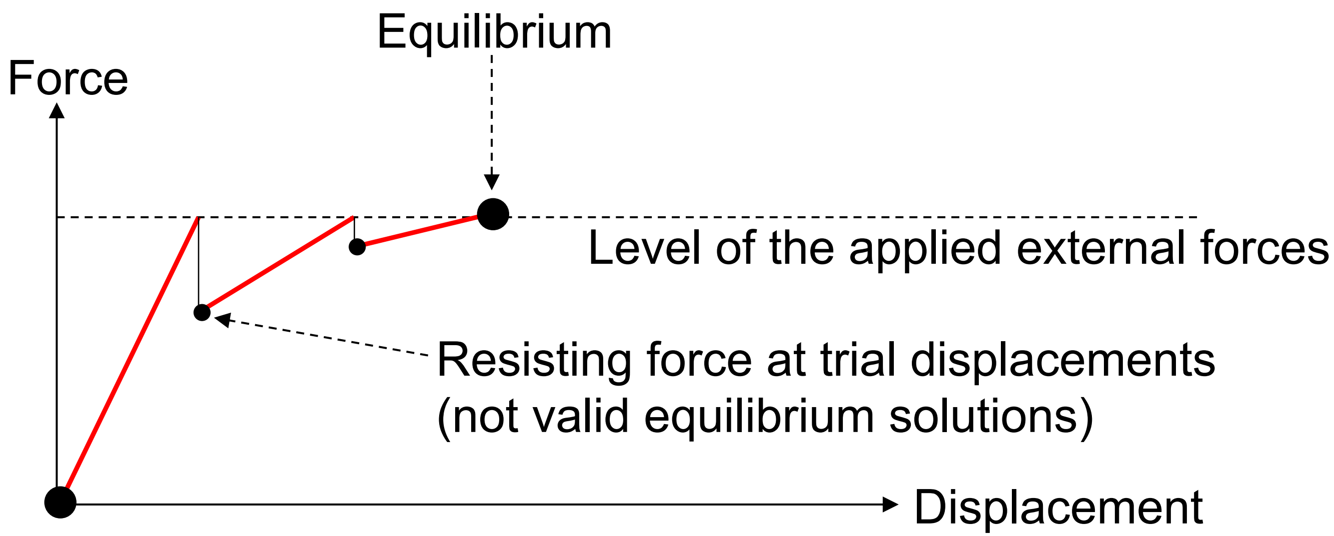
Linear system of equations in each iteration: $\frac{\partial \mathbf{R}(\mathbf{u}_i)}{\partial \mathbf{u}} \Delta \mathbf{u} = -\mathbf{R}(\mathbf{u}_i)$

Newton-Raphson: $\mathbf{K}_{\text{tangent}} \, \Delta \mathbf{u} = -\mathbf{R}(\mathbf{u}_i)$

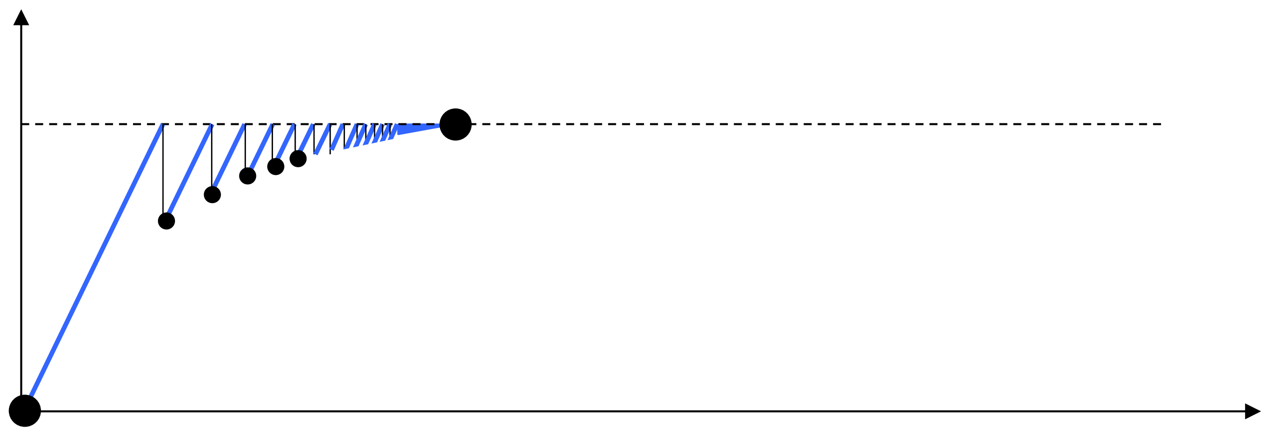
Modified Newton-Raphson: $\mathbf{K}_{\text{initial}} \, \Delta \mathbf{u} = -\mathbf{R}(\mathbf{u}_i)$

The Two Options

Newton-Raphson:

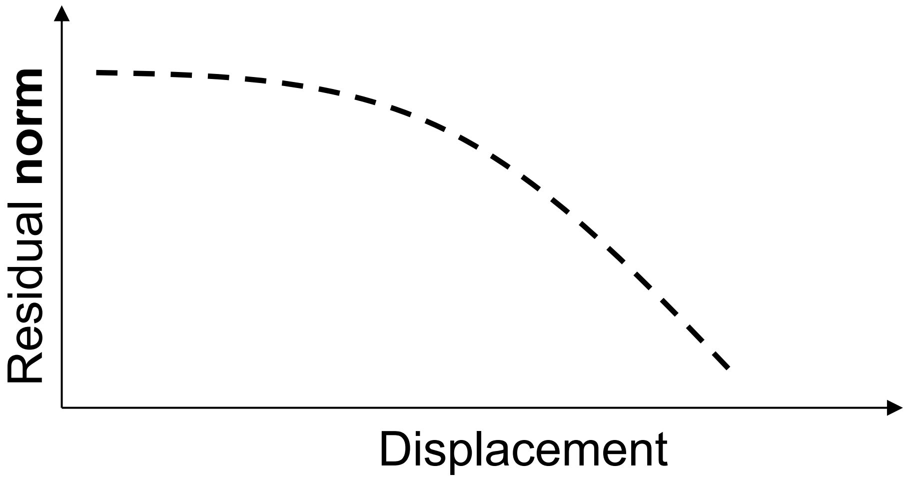


Modified Newton-Raphson:

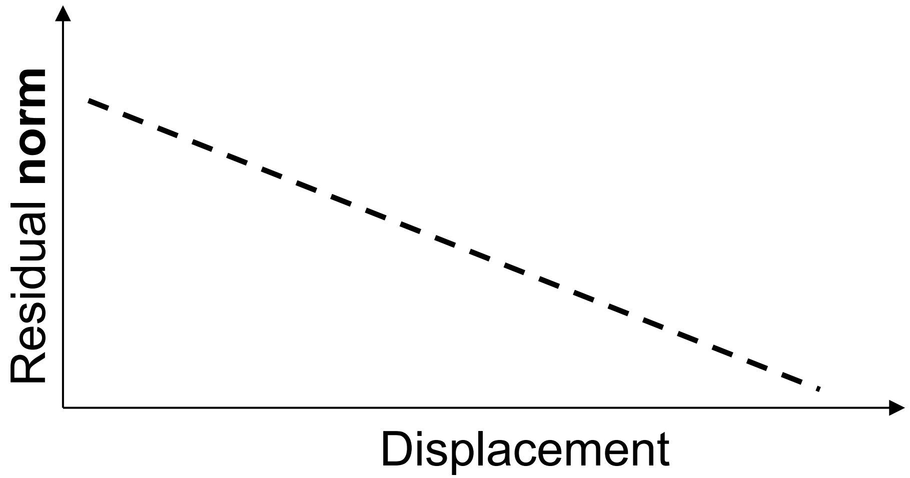


Convergence

Newton-Raphson:



Modified Newton-Raphson:



Python Code

```
while resNorm > tol and i < maxIterations:

    # Check if user wants Modified Newton-Raphson
    if m == stiffnessCalcFrequency:

        m = 0

        # Basic stiffnesses
        Kb1 = (spring1(ub1))[1]
        Kb2 = (spring2(ub2))[1]
        Kb3 = (spring3(ub3))[1]

        # Final stiffness matrix
        Kf = np.transpose(Tbf1 * Kb1).dot(Tbf1) + np.transpose(Tbf2 * Kb2).dot(Tbf2) + np.transpose(Tbf3 * Kb3).dot(Tbf3)

    # Solve for the displacement increment
    duf = np.linalg.solve(Kf, -Rf)

    # New trial displacements
    uf = uf + duf

    # State determination, starting with Basic displacements
    ub1 = np.dot(Tbf1, uf)
    ub2 = np.dot(Tbf2, uf)
    ub3 = np.dot(Tbf3, uf)

    # Basic forces
    Fb1 = (spring1(ub1))[0]
    Fb2 = (spring2(ub2))[0]
    Fb3 = (spring3(ub3))[0]

    # Final force vector
    tildeFf = np.dot(Tbf1.transpose(), Fb1) + np.dot(Tbf2.transpose(), Fb2) + np.dot(Tbf3.transpose(), Fb3)

    # Residual vectdor and its norm
    Rf = tildeFf - Ff
    resNorm = np.linalg.norm(Rf)
```

State Determination

Trial displacements

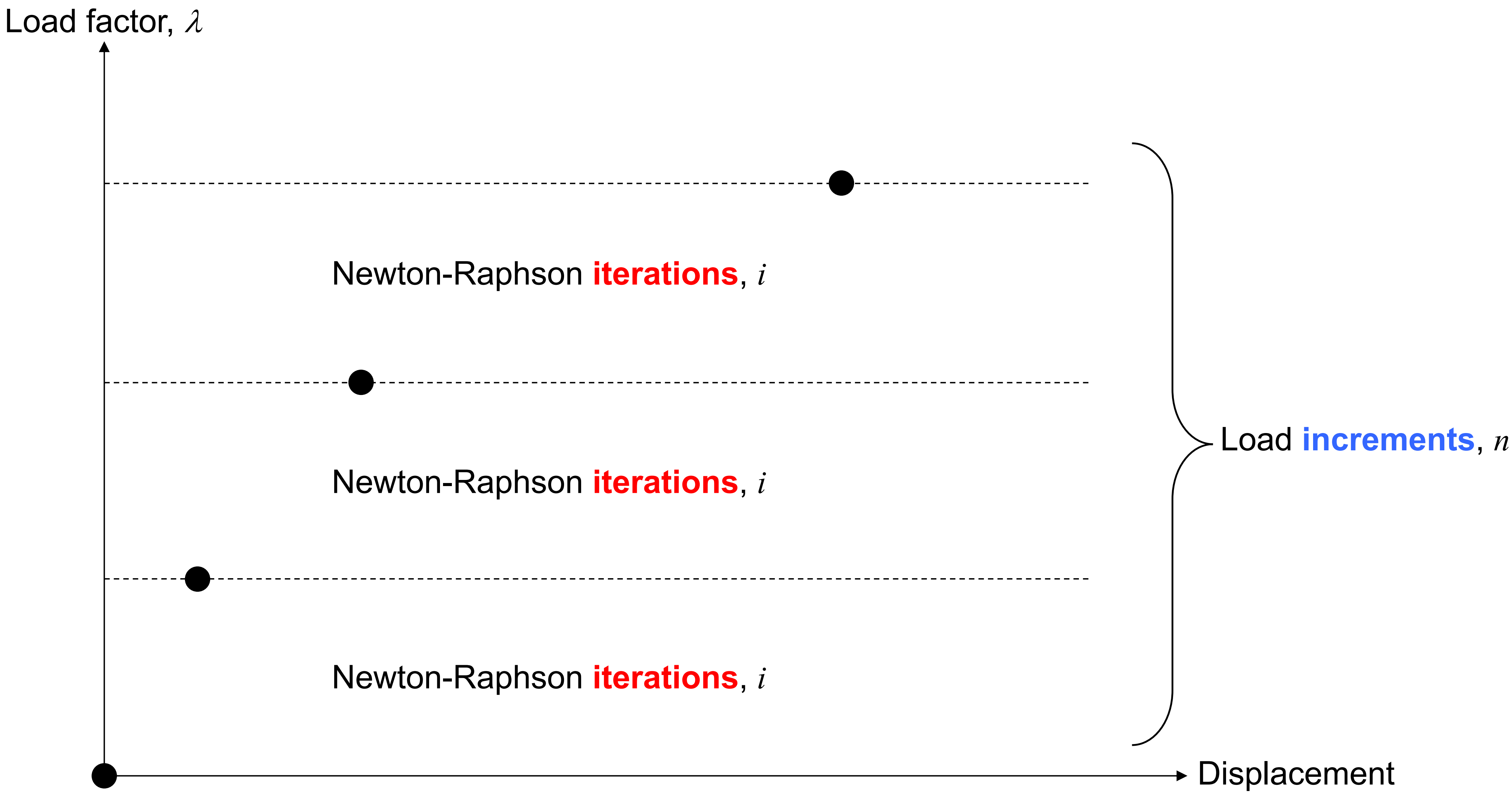
$$\varepsilon = \mathbf{T}_{ms}\mathbf{T}_{sb}\mathbf{T}_{bl}\mathbf{T}_{lg}\mathbf{T}_{ga}\mathbf{T}_{af}\mathbf{u}_f$$

Material model(s) that take total strain, or incremental strain, or both

$$\tilde{\mathbf{F}}_f\left(\mathbf{u}_f\right) = \mathbf{T}_{af}^T \sum \left(\mathbf{T}_{ga}^T \mathbf{T}_{lg}^T \mathbf{T}_{bl}^T \left(\int_0^L \mathbf{T}_{sb}^{\mathrm{T}} \left(\int \mathbf{T}_{ms}^{\mathrm{T}} \cdot \sigma \, dA \right) dx + \bar{\mathbf{F}}_b \right) \right)$$

Check convergence

Iterations vs. Increments

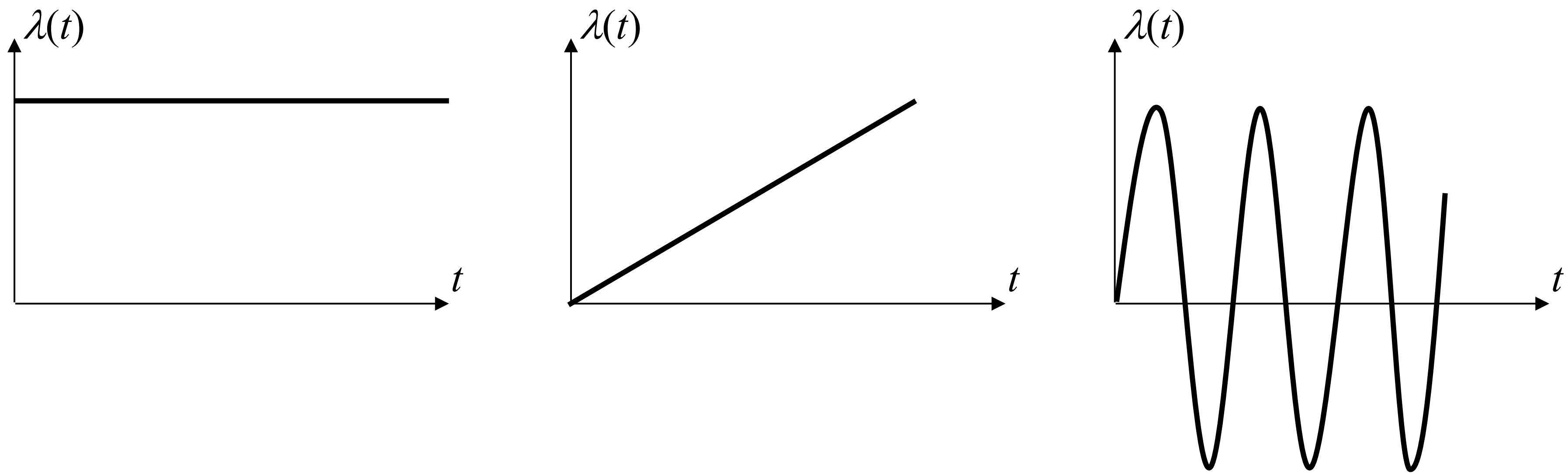


Load Factor, λ

Time series

$$\mathbf{F}_f(t) = \lambda(t) \cdot \mathbf{F}_{ref}$$

Load factor | Pseudo-time | Load pattern



Match Number of Time Steps with Δt

Suppose a 5kN load is to be applied gradually to the structure

Many options!

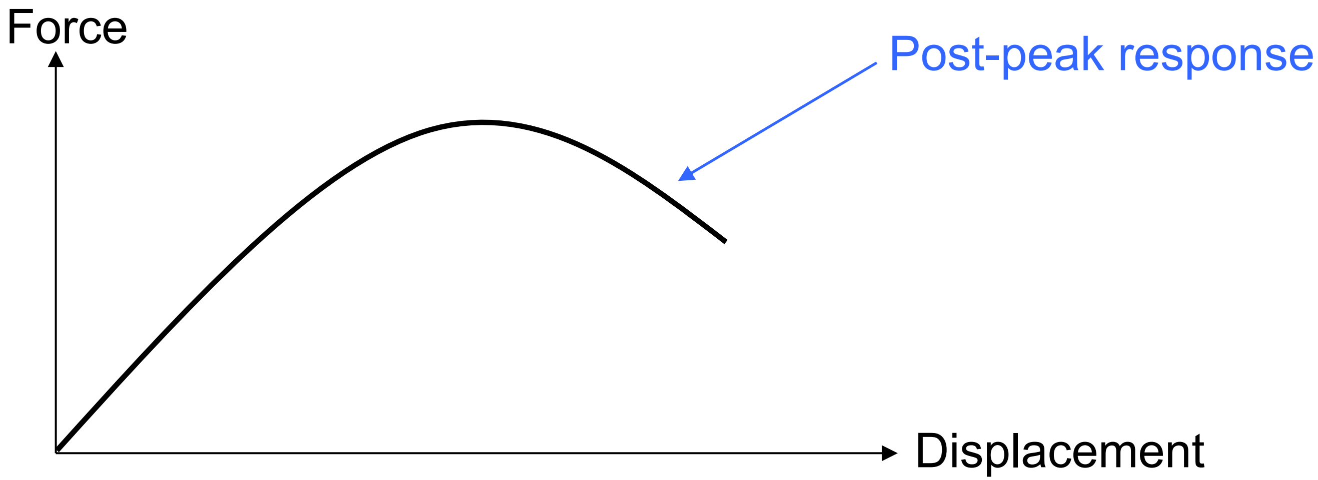
Set $\mathbf{F}_{\text{ref}} = 5\text{kN}$ and $\lambda(t)=t$, reaching the full load at pseudo time $t=1$

Set $\mathbf{F}_{\text{ref}} = 5\text{kN}$ and $\lambda(t)=0.01t$, reaching the full load at time $t=100$

Set $\mathbf{F}_{\text{ref}} = 1\text{kN}$ and $\lambda(t)=t$, reaching the full load at time $t=5$

Set $\mathbf{F}_{\text{ref}} = 1\text{kN}$ and $\lambda(t)=0.1t$, reaching the full load at time $t=50$

Continuation Methods



Load Control during Iterations, Part I

Isolate the change in the load factor in this increment: $\lambda_n = \lambda_{n-1} + \Delta\lambda$

Resulting increment in the load: $\mathbf{F}_n = \mathbf{F}_{n-1} + \Delta\lambda \cdot \mathbf{F}_{ref}$

Resulting expression for the residual: $\mathbf{R}_i = \tilde{\mathbf{F}}_i - \mathbf{F}_n = \tilde{\mathbf{F}}_i - \mathbf{F}_{n-1} - \Delta\lambda \cdot \mathbf{F}_{ref}$

Allow the load factor to vary within the iterations ($n \rightarrow i$): $\mathbf{R}_i = \tilde{\mathbf{F}}_i - \mathbf{F}_{i-1} - \Delta\lambda_i \cdot \mathbf{F}_{ref}$

Steadily accumulating trial displacements: $\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u}_i$

Linear system of equations for Δu_i: $\mathbf{K}\,\Delta\mathbf{u}_i = -\mathbf{R}_i$

Substitute expression for the residual: $\mathbf{K}\,\Delta\mathbf{u}_i = \mathbf{F}_{i-1} + \Delta\lambda_i \cdot \mathbf{F}_{ref} - \tilde{\mathbf{F}}_i$

Load Control during Iterations, Part II

First term in the split $\Delta\mathbf{u}_i = \Delta\mathbf{u}_{R,i} + \Delta\mathbf{u}_{T,i}$: $\mathbf{K}\,\Delta\mathbf{u}_{R,i} = \mathbf{F}_{i-1} - \tilde{\mathbf{F}}_i$

Namely: $\mathbf{K}\,\Delta\mathbf{u}_{R,i} = \lambda_{i-1} \cdot \mathbf{F}_{ref} - \tilde{\mathbf{F}}_i$

Second term in the split $\Delta\mathbf{u}_i = \Delta\mathbf{u}_{R,i} + \Delta\mathbf{u}_{T,i}$: $\mathbf{K}\,\Delta\mathbf{u}_{T,i} = \Delta\lambda_i \cdot \mathbf{F}_{ref}$

Define reference displacement, constant through iterations: $\mathbf{K}\,\mathbf{u}_T = \mathbf{F}_{ref}$

That means the second term is: $\Delta\mathbf{u}_{T,i} = \Delta\lambda_i \cdot \Delta\mathbf{u}_T$

So the split $\Delta\mathbf{u}_i$ reads: $\Delta\mathbf{u}_i = \Delta\mathbf{u}_{R,i} + \Delta\lambda_i \cdot \mathbf{u}_T$

Displacement Control

Enforce displacement at a control-DOF: One less unknown!

Selection vector: $\mathbf{s} = \{0,0,0,1,0,0,0,0,0\}$

Displacement increment at control-DOF: $\mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_i = \mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_{\mathrm{R},i} + \Delta\lambda_i \cdot \mathbf{s}^{\mathrm{T}}\mathbf{u}_{\mathrm{T}}$

First iteration: $\mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_i = \mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_{\mathrm{R},i} + \Delta\lambda_i \cdot \mathbf{s}^{\mathrm{T}}\mathbf{u}_{\mathrm{T}} = \Delta u_o$

Solve for $\Delta\lambda_i$, remembering that $\Delta\mathbf{u}_{\mathrm{R},i}$ is zero at first iteration: $\Delta\lambda_1 = \frac{\Delta u_o}{\mathbf{s}^{\mathrm{T}}\mathbf{u}_{\mathrm{T}}}$

Later iterations: $\mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_i = \mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_{\mathrm{R},i} + \Delta\lambda_i \cdot \mathbf{s}^{\mathrm{T}}\mathbf{u}_{\mathrm{T}} = 0$

Result: $\Delta\lambda_i = -\frac{\mathbf{s}^{\mathrm{T}}\Delta\mathbf{u}_{\mathrm{R},i}}{\mathbf{s}^{\mathrm{T}}\mathbf{u}_{\mathrm{T}}}$

More lectures:

Terje's Toobox:

terje.civil.ubc.ca