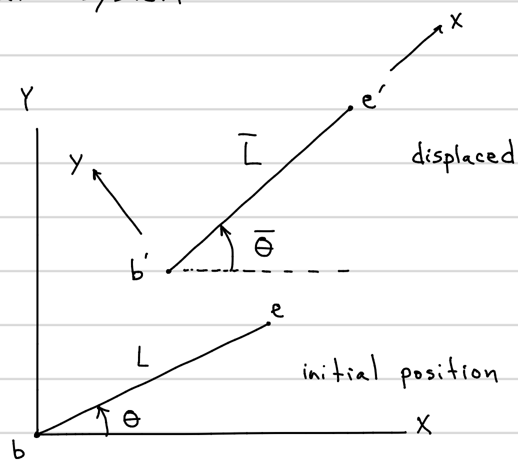
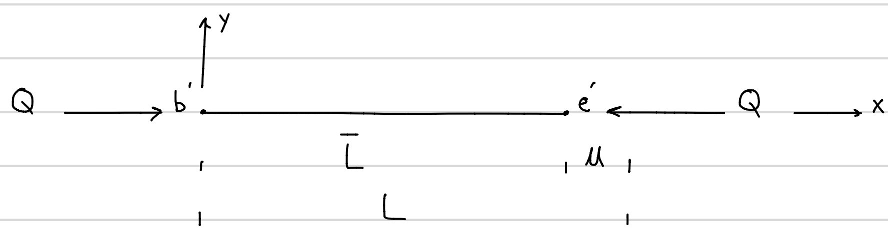
November 26, 2024
Matrix Displacement Method
Nonlinear Analysis (geometric)
Part II - Plane Truss

Geometric nonlinear analysis of plane trusses

Unlike linear analysis where local coordinate system is positioned in the undeformed state, in geometrically nonlinear analysis the local coordinate system is attatched to, and translates/rotates, with the member as the structure deforms : corotational (Eulerian) coordinate system



In Eulerian coordinate system, only one DOF (axial deformation - u) is needed to completely specify displaced position of member



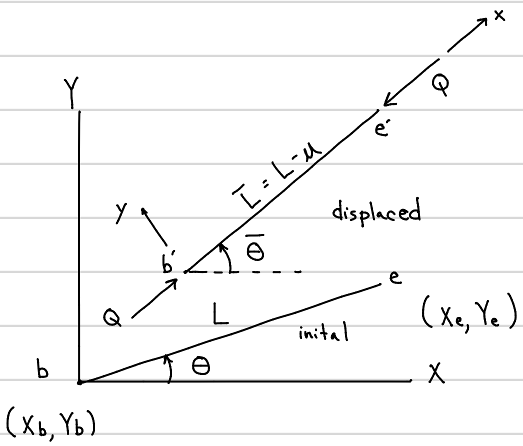
$u = L - \bar{L}$
(positive when compressed)

recall $\sigma = \dfrac{Q}{A}$ $\qquad \varepsilon = \dfrac{u}{L}$ $\qquad$ constitutive relation $\sigma = E\varepsilon$ (linear elastic material)

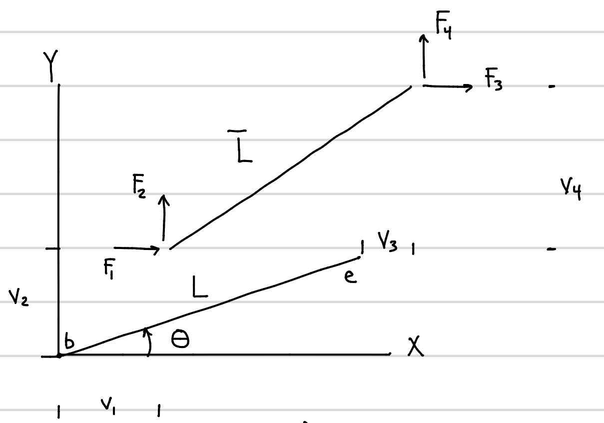
combining relations $\quad Q = \left(\dfrac{EA}{L}\right) u$

* $\varepsilon = \dfrac{L - \bar{L}}{L}$ nonlinear since $\bar{L}$ is NOT a linear combination of member-end displacements *

$\left(\varepsilon_{prior} = \dfrac{u_2 - u_1}{L}\right)$



(local)



(global)

$$L = \sqrt{(X_e - X_b)^2 + (Y_e - Y_b)^2}$$

$$\bar{L} = \sqrt{\left[(X_e + v_3) - (X_b + v_1)\right]^2 + \left[(Y_e + v_4) - (Y_b + v_2)\right]^2}$$

$$C_X = \cos\bar{\theta} = \frac{(X_e + v_3) - (X_b + v_1)}{\bar{L}} = \frac{L_{\bar{x}}}{\bar{L}}$$

$$C_Y = \sin\bar{\theta} = \frac{(Y_e + v_4) - (Y_b + v_2)}{\bar{L}} = \frac{L_{\bar{y}}}{\bar{L}}$$

$$F_1 = C_X Q \qquad F_2 = C_Y Q \qquad F_3 = -C_X Q \qquad F_4 = -C_Y Q$$

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{Bmatrix} C_X \\ C_Y \\ -C_X \\ -C_Y \end{Bmatrix} Q \qquad \{F\} = [T]^T \{Q\}$$

$$[T] = \begin{bmatrix} C_x & C_y & -C_x & -C_Y \end{bmatrix}$$

Note: $\{F\}$ is in terms of $\{v\}$ $\therefore$ nonlinear (geometrically exact)

If deformed configuration (displacements) known, calculation of joint forces can be determined by direct application of nonlinear force-displacement relations previously derived (displacement-controlled loading)

$$\{F\} = [T]^T \{Q\} \qquad [T] = [C_X \;\; C_Y \;\; -C_X \;\; -C_Y] \qquad Q = \left(\frac{EA}{L}\right)(L - \bar{L})$$

However, if external forces are specified and displacements need to be determined, this requires solving system of nonlinear equations via iterative techniques, e.g. Newton Raphson Method (force-controlled loading)

Member Tangent Stiffness Matrix $[K_t]$

$$[K_t] = \left[\frac{\partial F_i}{\partial v_j}\right] \quad \text{for } i,j = 1\text{-}4$$

$$\underset{4\times4}{[K_t]} = \begin{bmatrix} \frac{\partial F_1}{\partial v_1} & \frac{\partial F_1}{\partial v_2} & \frac{\partial F_1}{\partial v_3} & \frac{\partial F_1}{\partial v_4} \\ \frac{\partial F_2}{\partial v_1} & & & \\ \frac{\partial F_3}{\partial v_1} & & \text{etc.} & \\ \frac{\partial F_4}{\partial v_1} & & & \end{bmatrix}$$

calculating components, e.g. $\frac{\partial F_1}{\partial v_1}$

$F_1 = C_X Q$ both C_X and Q are functions of v_1 $\therefore$

$$\frac{\partial F_1}{\partial v_1} = C_X\left(\frac{\partial Q}{\partial v_1}\right) + Q\left(\frac{\partial C_X}{\partial v_1}\right) \quad \text{(chain rule)} \qquad \frac{\partial Q}{\partial v_1} = \left(\frac{EA}{L}\right) C_X \qquad \frac{\partial C_X}{\partial v_1} = \frac{-C_Y^2}{\bar{L}}$$

thus, $$\frac{\partial F_1}{\partial v_1} = \left(\frac{EA}{L}\right) C_X^2 - \left(\frac{Q}{\bar{L}}\right) C_Y^2$$

similarly, partial derivatives of remaining entries can be calculated

$$[K_t] = \left(\frac{EA}{L}\right)\begin{bmatrix} C_x^2 & C_xC_Y & -C_x^2 & -C_xC_Y \\ C_xC_Y & C_Y^2 & -C_xC_Y & -C_Y^2 \\ -C_x^2 & -C_xC_Y & C_x^2 & C_xC_Y \\ -C_xC_Y & -C_Y^2 & C_xC_Y & C_Y^2 \end{bmatrix} + \left(\frac{Q}{L}\right)\begin{bmatrix} -C_Y^2 & C_xC_Y & C_Y^2 & -C_xC_Y \\ C_xC_Y & -C_x^2 & -C_xC_Y & C_x^2 \\ C_Y^2 & -C_xC_Y & -C_Y^2 & C_xC_Y \\ -C_xC_Y & C_x^2 & C_xC_Y & -C_x^2 \end{bmatrix}$$

in compact form

$$[K_t] = \frac{EA}{L}[T]^T[T] + Q\underbrace{[G]}_{\text{geometric matrix}}$$

Solving Structural System of nonlinear equations : $P = f(d)$
(assembly via code number, e.g. $F \rightarrow f$)

$P = f(d)$, i.e. $P^{ext} = f^{int}$ equilibrium attained when $P^{ext} - f^{int} \approx 0$

iterative (i) Newton Raphson Technique

$$\{d\}_{i+1} = \{d\}_i + \{\Delta d\}_i$$

$$\{\Delta d\}_i = [S_t]_i^{-1}\left(\{P^{ext}\} - \{f^{int}(d_i)\}\right)$$

$\{P^{ext}\} - \{f^{int}(d_i)\} \rightarrow \{\Delta U\}_i$ unbalanced force vector

$[S_t]_i \rightarrow [S_t(d_i)]$ assembled from $[K_t]$

Convergence achieved when $R_i = \|\{P^{ext} - f^{int}\}_i\| < \varepsilon R_o$

* alternate convergence criteria exist (Δd next page)

L2 (Euclidean) norm : $\|X\| = \sqrt{\sum_{k=1}^{n} X_k^2}$

Quadratic convergence achieved if exact $[K_t]$ implemented in each iteration

$$\lim_{k\to\infty} \frac{|X_{k+1} - X_{exact}|}{|X_k - X_{exact}|} = r = 2$$

Modified Newton Raphson uses $[K_t]_o$ throughout each load step, thus lowering inversion/factorization cost, but sub-quadratic convergence and increased iterations

Nonlinear computer code flowchart