FEM advanced course

Lecture 7 - Geometrically exact Reissner beam formulation and path following algorithms

Reijo Kouhia

Tampere University, Structural Mechanics

Geometrically exact Reissner beam model - kinematics (1/4)

It is a geometrically non-linear formulation of the Timoshenko beam model where the average transverse shear strain is taken into account. It is based on the following two kinematic assumptions:

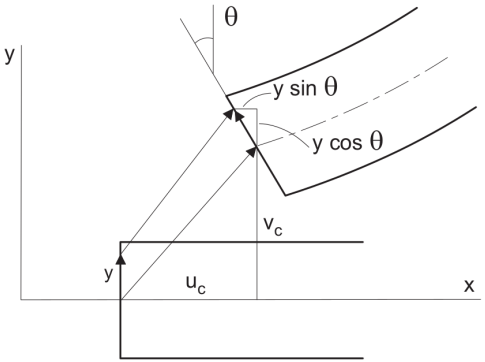
1. fibers normal to the beam's axis in the undeformed state remain straight,
2. these normal fibers do not stretch.

These assumptions can be satisfied by the following displacement field

$$u(X_1, X_2) = u_c(X_1) - X_2 \sin\theta(X_1),$$
$$v(X_1, X_2) = v_c(X_1) - X_2(1 - \cos\theta(X_1)).$$

Deformation mapping is thus

$$\varphi_1(X_1, X_2) = X_1 + u_c(X_1) - X_2 \sin\theta(X_1),$$
$$\varphi_2(X_1, X_2) = X_2 + v_c(X_1) - X_2(1 - \cos\theta(X_1)$$
$$= v_c(X_1) + X_2 \cos\theta(X_1)$$



Geometrically exact Reissner beam model - kinematics (2/4)

Deformation gradient is

$$\boldsymbol{F} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{X}} = \begin{bmatrix} 1 + u_c' - X_2(\cos\theta)\theta' & -\sin\theta \\ v_c' - X_2(\sin\theta)\theta' & \cos\theta \end{bmatrix}$$

Polar decomposition is $\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$, in plane the right Cauchy-Green stretch tensor $\boldsymbol{U}$ can be obtained by a simple formula

$$\boldsymbol{U} = \frac{1}{\sqrt{I_1(\boldsymbol{C}) + 2J(\boldsymbol{C})}} \left(\boldsymbol{C} + J(\boldsymbol{C})\boldsymbol{I}\right),$$

where $I_1(\boldsymbol{C}) = C_{11} + C_{22}$ and $J(\boldsymbol{C}) = \sqrt{C_{11}C_{22}}$.

A more versatile decomposition is

$$\boldsymbol{Q}\,\tilde{\boldsymbol{U}}, \quad \text{where} \quad \boldsymbol{Q} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Then

$$\tilde{\boldsymbol{U}} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{F} = \begin{bmatrix} (1 + u_c')\cos\theta + v_c'\sin\theta - X_2\theta' & 0 \\ -(1 + u_c')\sin\theta + v_c'\cos\theta & 1 \end{bmatrix}$$

Notice that $\tilde{\boldsymbol{U}}$ is not symmetric.

Geometrically exact Reissner beam model - kinematics (3/4)

Biot type strain measure is now defined as

$$\begin{aligned}\tilde{\boldsymbol{E}}^{(1)} &= \tilde{\boldsymbol{U}} - \boldsymbol{I} \\ &= \begin{bmatrix} (1+u_c')\cos\theta + v_c'\sin\theta - X_2\theta' & 0 \\ -(1+u_c')\sin\theta + v_c'\cos\theta & 0 \end{bmatrix} \\ &= \begin{bmatrix} \varepsilon_c + X_2\kappa & 0 \\ \gamma & 0 \end{bmatrix},\end{aligned}$$

where

$$\begin{aligned}\varepsilon_c &= (1+u_c')\cos\theta + v_c'\sin\theta, \\ \kappa &= -\theta', \\ \gamma &= -(1+u_c')\sin\theta + v_c'\cos\theta.\end{aligned}$$

Geometrically exact Reissner beam model - kinematics (4/4)

The strain $\tilde{\boldsymbol{E}}^{(1)}$ is conveniently displayed in a vector form

$$\left\{e^{(1)}\right\} = [T(\theta)]\,\{u'\} - \{N\}\,,$$

where

$$\left\{e^{(1)}\right\} = \begin{Bmatrix} \varepsilon_c \\ \gamma \\ \kappa \end{Bmatrix} \quad [T(\theta)] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \{u'\} = \begin{Bmatrix} 1+u_c' \\ v_c' \\ -\theta' \end{Bmatrix}, \quad \{N\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}.$$

In 3D the problem is much more difficult. An excellent paper is J. Mäkinen, Total Lagrangian Reissner's geometrically exact beam element without singularities, *International journal for numerical methods in engineering*, vol 70, issue 9, 2006 https://doi.org/10.1002/nme.1892

Geometrically exact Reissner beam model - virtual work

The weak form is

$$-\int_0^L \left(N\delta\varepsilon_c + Q\delta\gamma + M\delta\kappa\right) \mathrm{d}X_1 + \delta W^{\mathrm{ext}} = 0.$$

The resultant form constitutive equations are

$$\begin{aligned} N &= EA\varepsilon_c, \\ Q &= GA_{\mathrm{s}}\gamma, \\ M &= EI\kappa. \end{aligned}$$

The shear rigidity should be redefined as

$$GA_{\mathrm{s}} \rightarrow \frac{GA_{\mathrm{s}}}{1 + \dfrac{GA_{\mathrm{s}}(h^{(e)})^2}{12EI}}.$$

to avoild ill-conditioning of the stiffness matrix.

Solution of parametrized non-linear algebraic equations (1/8)

The FE discretization produces non-linear algebraic equation system

$$\boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{r}(\boldsymbol{q}) - \boldsymbol{p}(\boldsymbol{q}),$$

where $\boldsymbol{r}$ is the internal resistance force vector, $\boldsymbol{p}$ is the external load vector which may or may not depend on displacement vector $\boldsymbol{q}$.

In many cases the load can be parametrized with a single non-dimensional variable, the load parameter λ as $\boldsymbol{p} = \lambda \boldsymbol{p}_{\mathrm{r}}$, where $\boldsymbol{p}_{\mathrm{r}}$ is a reference external force vector.

The equilibrium equation is then

$$\boldsymbol{f}(\boldsymbol{q}, \lambda) = \boldsymbol{r}(\boldsymbol{q}) - \lambda \boldsymbol{p}_{\mathrm{r}}(\boldsymbol{q}). \tag{1}$$

If the loads does not dependent on deformations, like in dead-weight loading, the reference load vector $\boldsymbol{p}_{\mathrm{r}}$ is a constant vector.

If the dimension of the displacement vector $\boldsymbol{q}$ is N, then equation (1) define a one dimensional equilibrium curve in a $N+1$ dimensional displacement-load space.

Procedures to trace such a one dimensional equilibrium path are called **continuation** or path following methods. They are incremental, step-wise algorithms. A typical continuation step includes the predictor and the corrector phases.

Solution of parametrized non-linear algebraic equations (2/8)

Constant load incrementation (see lecture 1), solution of system $\boldsymbol{f}(\boldsymbol{q}, \lambda) = \boldsymbol{0}$

1. Select an initial value $\boldsymbol{q}_1^0$, usually a zero vector if $\lambda_0 = 0.$ and compute $r_0 = \|\boldsymbol{f}(\boldsymbol{q}_0^1)\|$
2. Increment load $\lambda_n = \lambda_{n-1} + \Delta\lambda$
3. Set $i = 0$, and $\Delta\boldsymbol{q}_n = \boldsymbol{0}$ and iterate until convergence
   - (i) Compute $\boldsymbol{f}'(\boldsymbol{q}_n^i)$
   - (ii) **Solve** $\boldsymbol{f}'(\boldsymbol{q}_n^i)\delta\boldsymbol{q} = -\boldsymbol{f}(\boldsymbol{q}_n^i)$
   - (iii) Update $\Delta\boldsymbol{q}_n^{i+1} = \Delta\boldsymbol{q}_n^i + \delta\boldsymbol{q}$
   - (iv) Update $\boldsymbol{q}_n^{i+1} = \boldsymbol{q}_{n-1} + \Delta\boldsymbol{q}_n^{i+1}$
   - (v) Set $i = i + 1$
   - (vi) Compute $\boldsymbol{f}(\boldsymbol{q}^i)$
   - (vii) If $\|\boldsymbol{f}(\boldsymbol{q}^i)\| < \epsilon_\mathrm{r} r_0 + \epsilon_\mathrm{a}$ and $\|\delta\boldsymbol{q}\| > \epsilon_\mathrm{r}\|\Delta\boldsymbol{q}^i\| + \epsilon_\mathrm{a}$ converged and proceed to a new step, go to 2.

Solution of parametrized non-linear algebraic equations (3/8)

Pseudo arc-length methods

If λ is also considered as an unknown, then in the system (1) there are N equations and $N+1$ unknowns.

A constraint equation $c(\boldsymbol{q}, \lambda)$ in the displacement-load space is needed.

$$\boldsymbol{h}(\boldsymbol{q}, \lambda) = \left\{ \begin{array}{rcl} \boldsymbol{f}(\boldsymbol{q}, \lambda) & = & \boldsymbol{0} \\ c(\boldsymbol{q}, \lambda) & = & 0. \end{array} \right. \tag{2}$$

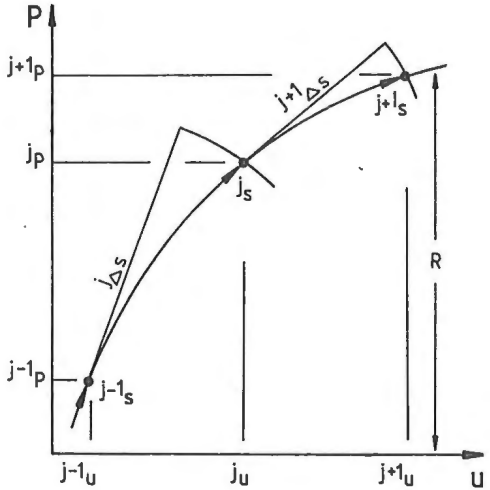
This kind of procedures are also commonly called arc-length methods. Using the Newton-Raphson linearization on the extended system (2) results in

$$\left\{ \begin{array}{rcl} \boldsymbol{f}'\delta\boldsymbol{q} + \boldsymbol{f}_{,\lambda}\delta\lambda + \boldsymbol{f}(\boldsymbol{q}, \lambda) & = & \boldsymbol{K}\delta\boldsymbol{q} - \boldsymbol{p}_r\delta\lambda + \boldsymbol{f} \quad = \quad \boldsymbol{0} \\ c'\delta\boldsymbol{q} + c_{,\lambda}\delta\lambda + c(\boldsymbol{q}, \lambda) & = & \boldsymbol{b}^T\delta\boldsymbol{q} + e\delta\lambda + c \quad = \quad 0 \end{array} \right. , \tag{3}$$

where $\boldsymbol{f}' = \partial \boldsymbol{f}/\partial \boldsymbol{q} = \boldsymbol{K}$ and $\boldsymbol{f}_{,\lambda} = \partial \boldsymbol{f}/\partial \lambda$, $c' = \partial c/\partial \boldsymbol{q} = \boldsymbol{b}^{\mathrm{T}}$ and $c_{,\lambda} = \partial c/\partial \lambda = e$.

Solution of parametrized non-linear algebraic equations (4/8)

Illustration in 1D case



Solution of parametrized non-linear algebraic equations (5/8)

Block elimination scheme

To utilize the spesific sparsity pattern and possible symmetry of the tangent stiffness matrix $\boldsymbol{K}$, the solution of the augmented equations (3) is usually performed by the three phase block elimination method, also known as bordering algorithm

1. solve $\boldsymbol{K}\delta\boldsymbol{q}_f = -\boldsymbol{f}$ and $\boldsymbol{K}\boldsymbol{q}_p = \boldsymbol{p}_r$,
2. compute $\delta\lambda = -(c + \boldsymbol{b}^T\delta\boldsymbol{q}_f)/(e + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}_p)$,
3. compute $\delta\boldsymbol{q} = \delta\boldsymbol{q}_f + \delta\lambda\boldsymbol{q}_p$.

If a direct linear solver is used the computational workload consist of

- one factorization of $\boldsymbol{K}$,
- two reductions of vectors $-\boldsymbol{f}$ and $\boldsymbol{p}_{\mathrm{r}}$ and backsubstitutions to obtain $\delta\boldsymbol{q}_f$ and $\boldsymbol{q}_p$,
- two scalar products and one vector update.

Solution of parametrized non-linear algebraic equations (6/8)

Nonsymmetric sparse format

$$\boldsymbol{H}\delta\boldsymbol{y} = -\boldsymbol{h}, \qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{p}_r \\ \boldsymbol{b}^T & e \end{bmatrix}, \qquad \delta\boldsymbol{y} = \begin{Bmatrix} \delta\boldsymbol{q} \\ \delta\lambda \end{Bmatrix}, \qquad \boldsymbol{h} = \begin{Bmatrix} \boldsymbol{f} \\ c \end{Bmatrix},$$

is more convenient if the linear system is solved with iterative methods, like Krylov subspace methods.

Solution of parametrized non-linear algebraic equations (7/8)

Forms of constraint

$$c(\boldsymbol{q}, \lambda) = \boldsymbol{t}^T \boldsymbol{C} \boldsymbol{n} - c_0 = 0 \tag{4}$$

where $\boldsymbol{t}$ and $\boldsymbol{n}$ are $n+1$ dimensional vectors and c_0 is a scalar. The weighting matrix $\boldsymbol{C}$ can be partitioned as

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{W} & \\ & \alpha^2 \end{bmatrix}, \tag{5}$$

where $\boldsymbol{W}$ is a positive definite or semidefinite diagonal matrix corresponding to displacements and α is a scaling factor. Updating the weight factors in $\boldsymbol{W}$ has proved to be beneficial for overall efficiency. Intuitively it can be understood easily, since then the process puts more weight on the most rapidly changing parts.

Solution of parametrized non-linear algebraic equations (8/8)

constraint	$\boldsymbol{t}^T$	$\boldsymbol{n}^T$	c_0	References
NP	$\boldsymbol{t}_1^T/\|\boldsymbol{t}_1\|_C$	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	Δs	E. Riks 1970, E. Ramm 1981
UNP	$\boldsymbol{t}_{i-1}^T/\|\boldsymbol{t}_{i-1}\|_C$	$[(\Delta\boldsymbol{q}_i)^T, \Delta\lambda_i]$	Δs	Ramm 1981
E	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	$(\Delta s)^2$	M. Crisfield 1981
VCP	$[(\Delta\boldsymbol{q}_i)^T, \Delta\lambda_i]$	$\boldsymbol{e}_k$	Δs	Rheinboldt 1977

NP	normal plane
UNP	updated normal plane
E	elliptical
VCP	variable control parameter
$\boldsymbol{t}_j^T$	$[\Delta\boldsymbol{q}_j^T, \Delta\lambda_j]$
Δ	incremental quantity
δ	iterative change
Δs	(pseudo) arc-length
$\boldsymbol{e}_k$	a unit vector having a component 1 at position k