Duke University
Department of Civil and Environmental Engineering
CEE 421L. Matrix Structural Analysis
Henri P. Gavin
Fall, 2012

3D Truss Analysis

1 Element Stiffness Matrix in Local Coordinates

Consider the relation between axial forces, $\{q_1, q_2\}$, and axial displacements, $\{u_1, u_2\}$, *only* (in local coordinates).

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

$$\mathbf{q} = \mathbf{k}\ \mathbf{u}$$

2 Coordinate Transformation

Global and local coordinates

.

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\cos\theta_x = \frac{x_2 - x_1}{L} = c_x$$

$$\cos\theta_y = \frac{y_2 - y_1}{L} = c_y$$

$$\cos\theta_z = \frac{z_2 - z_1}{L} = c_z$$

Displacements

.

$$u_1 = v_1 \cos\theta_x + v_2 \cos\theta_y + v_3 \cos\theta_z$$

$$u_2 = v_4 \cos\theta_x + v_5 \cos\theta_y + v_6 \cos\theta_z$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} c_x & c_y & c_z & 0 & 0 & 0 \\ 0 & 0 & 0 & c_x & c_y & c_z \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}$$

$$\mathbf{u} = \mathbf{T}\ \mathbf{v}$$

Forces

.

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} c_x & 0 \\ c_y & 0 \\ c_z & 0 \\ 0 & c_x \\ 0 & c_y \\ 0 & c_z \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$\mathbf{f} = \mathbf{T}^T\ \mathbf{q}$$

3 Element Stiffness Matrix in Global Coordinates

$$\left[\begin{array}{c} q_1 \\ q_2 \end{array}\right] = \frac{EA}{L} \left[\begin{array}{rr} 1 & -1 \\ -1 & 1 \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \end{array}\right] \qquad \mathbf{f} = \mathbf{T}^T \, \mathbf{q} \qquad \mathbf{u} = \mathbf{T} \, \mathbf{v}$$

$$\begin{aligned} \mathbf{q} &= \mathbf{k} \; \mathbf{u} \\ \mathbf{q} &= \mathbf{k} \; \mathbf{T} \; \mathbf{v} \\ \mathbf{T}^T \mathbf{q} &= \mathbf{T}^T \; \mathbf{k} \; \mathbf{T} \; \mathbf{v} \\ \mathbf{f} &= \mathbf{T}^T \; \mathbf{k} \; \mathbf{T} \; \mathbf{v} \\ \mathbf{f} &= \mathbf{K} \; \mathbf{v} \end{aligned}$$

$$\mathbf{K} = \frac{EA}{L} \left[\begin{array}{cccccc} c_x^2 & c_x c_y & c_x c_z & -c_x^2 & -c_x c_y & -c_x c_z \\ c_x c_y & c_y^2 & c_y c_z & -c_x c_y & -c_y^2 & -c_y c_z \\ c_x c_z & c_y c_z & c_z^2 & -c_x c_z & -c_y c_z & -c_z^2 \\ -c_x^2 & -c_x c_y & -c_x c_z & c_x^2 & c_x c_y & c_x c_z \\ -c_x c_y & -c_y^2 & -c_y c_z & c_x c_y & c_y^2 & c_y c_z \\ -c_x c_z & -c_y c_z & -c_z^2 & c_x c_z & c_y c_z & c_z^2 \end{array}\right]$$

4 Numbering Convention for Degrees of Freedom

```
g = [ 3*j1-2 ; 3*j1-1 ; 3*j1 ;  3*j2-2 ; 3*j2-1 ; 3*j2 ];
```

5 Truss Bar Tensions, T

$$T = q_2 = (\mathbf{kTv})_2 = \frac{EA}{L} \; \left(c_x(v_4 - v_1) + c_y(v_5 - v_2) + c_z(v_6 - v_3) \right)$$

6 Modifying truss_2d.m to truss_3d.m

- Copy `truss_2d.m` to `truss_3d.m` —

  ```
  function [D,R,T,L,Ks] = truss_3d(XYZ,JTS,RCT,EA,P,D)
  ```

  Modifications to the input arguments:

  – the joint location matrix `XYZ` has x, y, and z coordinates . . . a `3 x J` matrix;

  – the reaction matrix `RCT` has x, y, and z coordinates . . . a `3 x J` matrix;

  – the joint load matrix `P` has x, y, and z coordinates . . . a `3 x J` matrix;

  – the prescribed displacement matrix `D` has x, y, and z coordinates . . . a `3 x J` matrix;

  Modification to the computed output:

  – the computed deflections `D` will be the x, y, z displacements at each joint, returned as a `3 x J` matrix;

  – the computed reactions `R` will be the x, y, z forces at each joint with a reaction, returned as a `3 x J` matrix;

  Modifications to the program itself:

  – Change how `DoF` is computed;

  – Change `[Ks,L] = truss_assemble_2d(XY,JTS,EA);` to
  `[Ks,L] = truss_assemble_3d(XYZ,JTS,EA);`

  – Change `T = truss_forces_2d(XY,JTS,EA,Dv);` to
  `T = truss_forces_3d(XYZ,JTS,EA,D);`

  – Modify the section of code relating the joint displacement vector `Dv` to the joint displacement matrix `D` to account for the fact that there are three degrees of freedom per joint.

  – Change `plot` commands to `plot3` commands and change `XY` to `XYZ`.
  For example, change . . .
  `plot( XY(1,JTS(:,b)), XY(2,JTS(:,b)), '-g' )`
  . . . to . . .
  `plot3( XYZ(1,JTS(:,b)), XYZ(2,JTS(:,b)), '-g' )`
  Also change the `ax` variable to account for the `Z` dimension.

- Copy `truss_element_2d.m` to `truss_element_3d.m` —

```
function K = truss_element_3d(x1,y1,z1,x2,y2,z2,EA)
L =
cx =
cy =
cz =
K =
```

- Copy `truss_assemble_2d.m` to `truss_assemble_3d.m` —

```
function [Ks,L] = truss_assemble_3d(XYZ,JTS,EA)
DoF =
x1 =
y1 =
z1 =
x2 =
y2 =
z2 =
[K, L(b)] = truss_element_3d(x1,y1,z1,x2,y2,z2,EA(b) );
g =
```

- Copy `truss_forces_2d.m` to `truss_forces_3d.m` —

```
function T = truss_forces_3d(XYZ,JTS,EA,D)
x1 =
y1 =
z1 =
x2 =
y2 =
z2 =
L =
cx =
cy =
cz =
T(b) =
```