Stiffness matrix assembly of truss structures

This formulation assumes that the global ordering packs nodal displacements so that the components of each node's displacement vector are grouped together.

$$\mathbf{d}^e = \begin{bmatrix} u^e_{1x} & u^e_{1y} & u^e_{1z} & u^e_{2x} & u^e_{2y} & u^e_{2z} \end{bmatrix}^\mathsf{T} = \begin{bmatrix} \mathbf{u}^e_1 & \mathbf{u}^e_2 \end{bmatrix}^\mathsf{T} \tag{1}$$

The element stiffness matrix is then formed by a transformation from a coordinate system aligned with the element to the global coordinates.

$$\mathbf{K}^e = (\mathbf{R}^e)^\mathsf{T} \cdot \mathbf{K}'^e \cdot \mathbf{R}^e \quad \text{where} \quad \mathbf{K}'^e = \frac{E^e A^e}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2}$$

Here $\mathbf{K}'^e$ is the stiffness matrix relating the nodal displacements tangent to the element with forces tangent to the element. Therefore, the transformation matrix $\mathbf{R}^e$ must be able to convert global displacements of each node. Thus the element internal forces can be written in terms of a coordinate system aligned with the element as

$$\mathbf{f}'^e = \mathbf{K}'^e \mathbf{d}'^e = \mathbf{K}'^e \left(\mathbf{R}^e \mathbf{d}^e\right), \tag{3}$$

where

$$\mathbf{R}^e = \frac{1}{l^e} \begin{bmatrix} x^e_{21} & y^e_{21} & z^e_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & x^e_{21} & y^e_{21} & z^e_{21} \end{bmatrix}, \tag{4}$$

and where $x^e_{21} = x^e_2 - x^e_1$ is the difference between nodal coordinates, and $l^e = \|\mathbf{x}^e_{21}\|$ is the element length. Note that embedded in the rotation matrix is the direction along the element, i.e.,

$$\mathbf{t}^e = \frac{1}{l^e} \begin{bmatrix} x^e_{21} & y^e_{21} & z^e_{21} \end{bmatrix}. \tag{5}$$

The overall assembly procedure consists of a loop over the elements in the mesh, over which each element's stiffness matrix is constructed in local coordinates and then transformed into global coordinates. The element stiffness matrix in global coordinates is then scattered and summed into the global stiffness matrix for the whole system. An example MATLAB code for this process is given in Fig. 1

```
% Given
%   x: location of nodes; 3xN matrix
%   conn: connectivity of nodes; 2xNe matrix.
% Compute:
%   K: global stiffness matrix; NxN matrix.
K = zeros(N);
for c = conn
  % Gathers 3x2 matrix of element node coordinates.
  xe = x(:,c);
  % Computes 3x1 vector from node 1 to node 2 of element.
  dx = xe(:,2) - xe(:,1);
  Re = [dx',0,0,0; 0,0,0,dx']/norm(dx);
  % Stiffness matrix in element coordinates (2x2).
  Ke1 = Ee*Ae/norm(dx)*[1, -1; -1, 1];
  % Transformation of stiffness matrix to global coordinates (6x6).
  Ke = Re'*Ke1*Re;
  % The sctr vector maps local -> global degrees of freedom.
  sctr(1:2:6) = 3*c-2;  % Indices of x-component of displacement.
  sctr(2:2:6) = 3*c-1;  % Indices of y-component of displacement.
  sctr(3:2:6) = 3*c;    % Indices of z-component of displacement.
  K(sctr,sctr) = K(sctr,sctr) + Ke;
end
```

Figure 1 – Example MATLAB code to compute stiffness matrix from a 3D truss system.