# 20

# Implementation of One-Dimensional Elements

## TABLE OF CONTENTS

This Chapter begins Part III of the course. This Part deals with the computer implementation of the Finite Element Method for static analysis. It is organized in "bottom up" fashion. It begins with simple topics, such as programming of bar and beam elements, and gradually builds up toward more complex models and calculations.

Specific examples of this Chapter illustrate the programming of one-dimensional elements: bars and beams, using *Mathematica* as implementation language.

## §20.1. The Plane Bar Element

The two-node, prismatic, two-dimensional bar element was studied in Chapters 2-3 for modeling plane trusses. It is reproduced in Figure 20.1 for conveniency. It has two nodes and four degrees of freedom. The element node displacements and conjugate forces are

$$\mathbf{u}^e = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \end{bmatrix}, \quad \mathbf{f}^e = \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \end{bmatrix}. \tag{20.1}$$

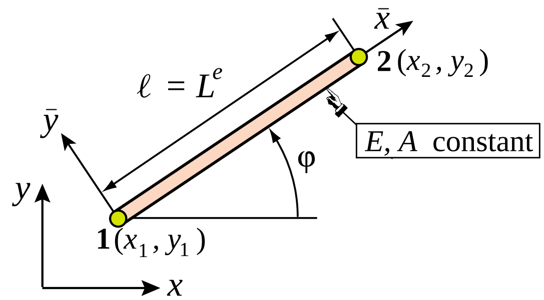


FIGURE 20.1. Plane bar element.

The element geometry is described by the coordinates $\{x_i, y_i\}$, $i = 1, 2$ of the two end nodes. For stiffness computations, the only material and fabrication properties required are the modulus of elasticity $E = E^e$ and the cross section area $A = A^e$, respectively. Both are taken to be constant over the element.

### §20.1.1. Stiffness Matrix

The element stiffness matrix in global $\{x, y\}$ coordinates is given by the explicit expression derived in §3.1:

$$\mathbf{K}^e = \frac{EA}{\ell} \begin{bmatrix} c^2 & sc & -c^2 & -sc \\ sc & s^2 & -sc & -s^2 \\ -c^2 & -sc & c^2 & sc \\ -sc & -s^2 & sc & s^2 \end{bmatrix} = \frac{EA}{\ell^3} \begin{bmatrix} x_{21}x_{21} & x_{21}y_{21} & -x_{21}x_{21} & -x_{21}y_{21} \\ x_{21}y_{21} & y_{21}y_{21} & -x_{21}y_{21} & -y_{21}y_{21} \\ -x_{21}x_{21} & -x_{21}y_{21} & x_{21}x_{21} & x_{21}y_{21} \\ -x_{21}y_{21} & -y_{21}y_{21} & x_{21}y_{21} & y_{21}y_{21} \end{bmatrix}. \tag{20.2}$$

Here $c = \cos\varphi = x_{21}/\ell$, $s = \sin\varphi = y_{21}/\ell$, in which $x_{21} = x_2 - x_1$, $y_{21} = y_2 - y_1$, $\ell = \sqrt{x_{21}^2 + y_{21}^2}$, and $\varphi$ is the angle formed by $\bar{x}$ and $x$, measured from $x$ positive counterclockwise (see Figure 20.1). The second expression in (20.2) is preferable in a computer algebra system because it enhances simplification possibilities when doing symbolic work, and is the one actually implemented in the module described below.

### §20.1.2. Stiffness Module

The computation of the stiffness matrix $\mathbf{K}^e$ of the two-node, prismatic plane bar element is done by *Mathematica* module `PlaneBar2Stiffness`. This is listed in Figure 20.2. The module is invoked as

```
Ke = PlaneBar2Stiffness[ncoor, Em, A, options]
```
(20.3)

```
PlaneBar2Stiffness[ncoor_,Em_,A_,options_]:= Module[
 {x1,x2,y1,y2,x21,y21,EA,numer,L,LL,LLL,Ke},
  {{x1,y1},{x2,y2}}=ncoor; {x21,y21}={x2-x1,y2-y1};
  EA=Em*A; {numer}=options; LL=x21^2+y21^2; L=Sqrt[LL];
  If [numer,{x21,y21,EA,LL,L}=N[{x21,y21,EA,LL,L}]];
  If [!numer, L=PowerExpand[L]]; LLL=Simplify[LL*L];
  Ke=(Em*A/LLL)*{{ x21*x21, x21*y21,-x21*x21,-x21*y21},
                 { y21*x21, y21*y21,-y21*x21,-y21*y21},
                 {-x21*x21,-x21*y21, x21*x21, x21*y21},
                 {-y21*x21,-y21*y21, y21*x21, y21*y21}};
  Return[Ke]];
```

FIGURE 20.2. *Mathematica* stiffness module for a two-node, prismatic plane bar element.

The arguments are

| | |
|---|---|
| `ncoor` | Node coordinates of element arranged as `{ { x1,y1 },{ x2,y2 } }`. |
| `Em` | Elastic modulus. |
| `A` | Cross section area. |
| `options` | A list of processing options. For this element is has only one entry: `{ numer }`. This is a logical flag with the value `True` or `False`. If `True` the computations are carried out in floating-point arithmetic. If `False` symbolic processing is assumed. |

The module returns the $4 \times 4$ element stiffness matrix as function value.

```
ClearAll[A,Em,L];
ncoor={{0,0},{30,40}}; Em=1000; A=5;
Ke= PlaneBar2Stiffness[ncoor,Em,A,{True}];
Print["Numerical Elem Stiff Matrix: "];
Print[Ke//MatrixForm];
Print["Eigenvalues of Ke=",Chop[Eigenvalues[N[Ke]]]];
Print["Symmetry check=",Simplify[Chop[Transpose[Ke]-Ke]]];
```

Numerical Elem Stiff Matrix:

$$\begin{pmatrix} 36. & 48. & -36. & -48. \\ 48. & 64. & -48. & -64. \\ -36. & -48. & 36. & 48. \\ -48. & -64. & 48. & 64. \end{pmatrix}$$

Eigenvalues of Ke = {200., 0, 0, 0}

Symmetry check={{0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0}}

FIGURE 20.3. Test of plane bar stiffness module with numerical inputs.

### §20.1.3. Testing the Plane Bar Module

The modules are tested by the scripts listed in Figures 20.3 and 20.4. The script shown on the top of Figure 20.3 tests a numerically defined element with end nodes located at (0, 0) and (30, 40), with $E = 1000$, $A = 5$, and `numer` set to `True`. Executing the script produces the results listed in the bottom of that figure.

```
ClearAll[A,Em,L];
ncoor={{0,0},{L,0}};
Ke=  PlaneBar2Stiffness[ncoor,Em,A,{False}];
kfac=Em*A/L; Ke=Simplify[Ke/kfac];
Print["Symbolic Elem Stiff Matrix: "];
Print[kfac," ",Ke//MatrixForm];
Print["Eigenvalues of Ke=",kfac,"*",Eigenvalues[Ke]];
```

Symbolic Elem Stiff Matrix:

$$\frac{\mathrm{A\ Em}}{\mathrm{L}}\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Eigenvalues of Ke $= \frac{\mathrm{A\ Em}}{\mathrm{L}} * \{0, 0, 0, 2\}$

FIGURE 20.4. Test of plane bar stiffness module with symbolic inputs.

On return from PlaneBar2Stiffness, the stiffness matrix returned in `Ke` is printed. Its four eigenvalues are computed and printed. As expected three eigenvalues, which correspond to the three independent rigid body motions of the element, are zero. The remaining eigenvalue is positive and equal to $EA/\ell$. The symmetry of `Ke` is checked by printing $(\mathbf{K}^e)^T - \mathbf{K}^e$ upon simplification and chopping.

The script of Figure 20.4 tests a symbolically defined bar element with end nodes located at (0, 0) and $(L, 0)$, which is aligned with the $x$ axis. Properties $E$ and $A$ are kept symbolic. Executing the script shown in the top of Figure 20.4 produces the results shown in the bottom of that figure. One thing to be noticed is the use of the stiffness scaling factor $EA/\ell$, called `kfac` in the script. This is a symbolic quantity that can be extracted as factor of matrix $\mathbf{K}^e$. The effect is to clean up matrix and vector output, as can be observed in the printed results.

## §20.2. The Space Bar Element

To show how the previous implementation extends easily to three dimensions, this section describes the implementation of the space bar element.

The two-node, prismatic, three-dimensional bar element is pictured in Figure 20.5. It has two nodes and six degrees of freedom. The element node displacements and congugate forces are arranged as

$$\mathbf{u}^e = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{z1} \\ u_{x2} \\ u_{y2} \\ u_{z2} \end{bmatrix}, \quad \mathbf{f}^e = \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{z1} \\ f_{x2} \\ f_{y2} \\ f_{z2} \end{bmatrix}. \tag{20.4}$$

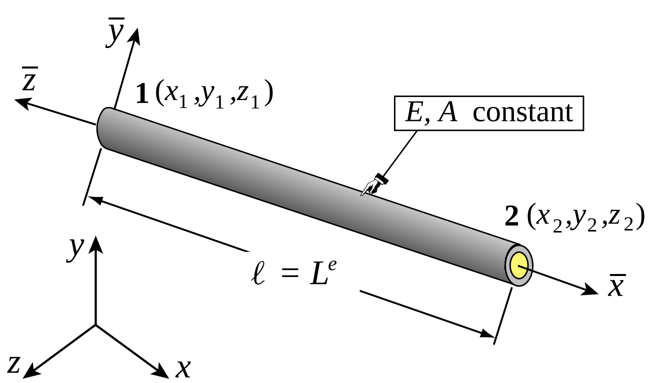


FIGURE 20.5. The space (3D) bar element.

The element geometry is described by the coordinates $\{x_i, y_i, z_i\}$, $i = 1, 2$ of the two end nodes. As in the case of the plane bar, the two properties required for the stiffness computations are the modulus of elasticity $E$ and the cross section area $A$. Both are assumed to be constant over the element.

```
SpaceBar2Stiffness[ncoor_,Em_,A_,options_]:=Module[
 {x1,x2,y1,y2,z1,z2,x21,y21,z21,EA,numer,L,LL,LLL,Ke},
  {{x1,y1,z1},{x2,y2,z2}}=ncoor;{x21,y21,z21}={x2-x1,y2-y1,z2-z1};
  EA=Em*A; {numer}=options;  LL=x21^2+y21^2+z21^2; L=Sqrt[LL];
  If [numer,{x21,y21,z21,EA,LL,L}=N[{x21,y21,z21,EA,LL,L}]];
  If [!numer, L=PowerExpand[L]]; LLL=Simplify[LL*L];
  Ke=(Em*A/LLL)*
     {{ x21*x21, x21*y21, x21*z21,-x21*x21,-x21*y21,-x21*z21},
      { y21*x21, y21*y21, y21*z21,-y21*x21,-y21*y21,-y21*z21},
      { z21*x21, z21*y21, z21*z21,-z21*x21,-z21*y21,-z21*z21},
      {-x21*x21,-x21*y21,-x21*z21, x21*x21, x21*y21, x21*z21},
      {-y21*x21,-y21*y21,-y21*z21, y21*x21, y21*y21, y21*z21},
      {-z21*x21,-z21*y21,-z21*z21, z21*x21, z21*y21, z21*z21}};
   Return[Ke];
];
```

FIGURE 20.6. Module to form the stiffness of the space (3D) bar element.

## §20.2.1. Stiffness Matrix

For the space bar element, introduce the notation $x_{21} = x_2 - x_1$, $y_{21} = y_2 - y_1$, $z_{21} = z_2 - z_1$ and $\ell = \sqrt{x_{21}^2 + y_{21}^2 + z_{21}^2}$. It can be shown[1] that the element stiffness matrix in global coordinates is given by

$$\mathbf{K}^e = \frac{E^e A^e}{\ell^3} \begin{bmatrix} x_{21}x_{21} & x_{21}y_{21} & x_{21}z_{21} & -x_{21}x_{21} & -x_{21}y_{21} & -x_{21}z_{21} \\ x_{21}y_{21} & y_{21}y_{21} & x_{21}z_{21} & -x_{21}y_{21} & -y_{21}y_{21} & -y_{21}z_{21} \\ x_{21}z_{21} & y_{21}z_{21} & z_{21}z_{21} & -x_{21}z_{21} & -y_{21}z_{21} & -z_{21}z_{21} \\ -x_{21}x_{21} & -x_{21}y_{21} & -x_{21}z_{21} & x_{21}x_{21} & x_{21}y_{21} & x_{21}z_{21} \\ -x_{21}y_{21} & -y_{21}y_{21} & -x_{21}z_{21} & x_{21}y_{21} & y_{21}y_{21} & y_{21}z_{21} \\ -x_{21}z_{21} & -y_{21}z_{21} & -z_{21}z_{21} & x_{21}z_{21} & y_{21}z_{21} & z_{21}z_{21} \end{bmatrix}. \tag{20.5}$$

This matrix expression in terms of coordinate differences is useful in symbolic work, because it enhances simplification possibilities.

## §20.2.2. Stiffness Module

The computation of the stiffness matrix $\mathbf{K}^e$ of the two-node, prismatic space bar element, is done by *Mathematica* module `SpaceBar2Stiffness`. This is listed in Figure 20.6. The module is invoked as

```
Ke = SpaceBar2Stiffness[ncoor, Em, A, options]
```
(20.6)

The arguments are

| | |
|---|---|
| `ncoor` | Node coordinates of element arranged as `{ { x1,y1,z1 },{ x2,y2,z2 } }`. |
| `Em` | Elastic modulus. |
| `A` | Cross section area. |
| `options` | A list of processing options. For this element is has only one entry: `{ numer }`. This is a logical flag with the value `True` or `False`. If `True` the computations are carried out in floating-point arithmetic. If `False` symbolic processing is assumed. |

The module returns the $6 \times 6$ element stiffness matrix as function value.

[1] The derivation was the subject of Exercise 6.10.

```
ClearAll[A,Em];
ncoor={{0,0,0},{2,3,6}}; Em=343; A=10;
Ke= SpaceBar2Stiffness[ncoor,Em,A,{True}];
Print["Numerical Elem Stiff Matrix: "];
Print[Ke//MatrixForm];
Print["Eigenvalues of Ke=",Chop[Eigenvalues[Ke]]];
```

Numerical Elem Stiff Matrix:

$$\begin{pmatrix} 40. & 60. & 120. & -40. & -60. & -120. \\ 60. & 90. & 180. & -60. & -90. & -180. \\ 120. & 180. & 360. & -120. & -180. & -360. \\ -40. & -60. & -120. & 40. & 60. & 120. \\ -60. & -90. & -180. & 60. & 90. & 180. \\ -120. & -180. & -360. & 120. & 180. & 360. \end{pmatrix}$$

Eigenvalues of Ke = {980., 0, 0, 0, 0, 0}

FIGURE 20.7. Testing the space bar stiffness module with numerical inputs.

```
ClearAll[A,Em,L];
ncoor={{0,0,0},{L,2*L,2*L}/3};
Ke=  SpaceBar2Stiffness[ncoor,Em,A,{False}];
kfac=Em*A/(9*L); Ke=Simplify[Ke/kfac];
Print["Symbolic Elem Stiff Matrix: "];
Print[kfac," ",Ke//MatrixForm];
Print["Eigenvalues of Ke=",kfac,"*",Eigenvalues[Ke]];
```

Symbolic Elem Stiff Matrix:

$$\frac{\text{A Em}}{9\text{ L}} \begin{pmatrix} 1 & 2 & 2 & -1 & -2 & -2 \\ 2 & 4 & 4 & -2 & -4 & -4 \\ 2 & 4 & 4 & -2 & -4 & -4 \\ -1 & -2 & -2 & 1 & 2 & 2 \\ -2 & -4 & -4 & 2 & 4 & 4 \\ -2 & -4 & -4 & 2 & 4 & 4 \end{pmatrix}$$

Eigenvalues of Ke = $\frac{\text{A Em}}{9\text{ L}} * \{0, 0, 0, 0, 0, 18\}$

FIGURE 20.8. Testing the space bar stiffness module with symbolic inputs.

## §20.2.3. Testing the Space Bar Module

The modules are tested by the scripts listed in Figures 20.7 and 20.8. As these are similar to previous tests done on the plane bar they need not be described in detail.

The script of Figure 20.7 tests a numerically defined space bar with end nodes located at (0, 0, 0) and (30, 40, 0), with $E = 1000$, $A = 5$ and `numer` set to `True`. Executing the script produces the results listed in the bottom of that Figure.

The script of Figure 20.8 tests a symbolically defined bar element with end nodes located at $(0, 0, 0)$ and $(L, 2L, 2L)/3$, which has length $L$ and is not aligned with the $x$ axis. The element properties $E$ and $A$ are kept symbolic. Executing the script produces the results shown in the bottom of that Figure. Note the use of a stiffness factor `kfac` of $EA/(9\ell)$ to get cleaner printouts.

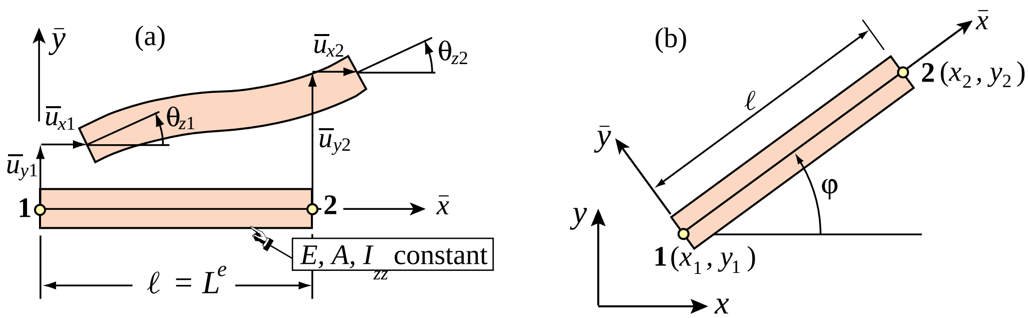


FIGURE 20.9. Plane beam-column element: (a) in its local system; (b) in the global system.

## §20.3. The Plane Beam-Column Element

Beam-column elements model structural members that resist both axial and bending actions. This is the case in skeletal structures such as frameworks which are common in steel and reinforced-concrete building construction. A plane beam-column element is a combination of a plane bar (such as that considered in §20.1), and a plane beam.

We consider a beam-column element in its local system $(\bar{x}, \bar{y})$ as shown in Figure 20.9(a), and then in the global system $(x, y)$ as shown in Figure 20.9(b). The six degrees of freedom and conjugate node forces of the elements are:

$$\bar{\mathbf{u}}^e = \begin{bmatrix} \bar{u}_{x1} \\ \bar{u}_{y1} \\ \theta_{z1} \\ \bar{u}_{x2} \\ \bar{u}_{y2} \\ \theta_{z2} \end{bmatrix}, \quad \bar{\mathbf{f}}^e = \begin{bmatrix} \bar{f}_{x1} \\ \bar{f}_{y1} \\ m_{z1} \\ \bar{u}_{x2} \\ \bar{u}_{y2} \\ m_{z2} \end{bmatrix}, \qquad \mathbf{u}^e = \begin{bmatrix} u_{x1} \\ u_{y1} \\ \theta_{z1} \\ u_{x2} \\ u_{y2} \\ \theta_{z2} \end{bmatrix}, \quad \mathbf{f}^e = \begin{bmatrix} f_{x1} \\ f_{y1} \\ m_{z1} \\ f_{x2} \\ f_{y2} \\ m_{z2} \end{bmatrix}. \tag{20.7}$$

The rotation angles $\theta$ and the nodal moments $m$ are the same in the local and the global systems because they are about the $z$ axis, which does not change in passing from local to global.

The element geometry is described by the coordinates $\{x_i, y_i\}$, $i = 1, 2$ of the two end nodes. The element length is $\ell = L^e$. Properties involved in the stiffness calculations are: the modulus of elasticity $E$, the cross section area $A$ and the moment of inertia $I = I_{zz}$ about the neutral axis. All properties are taken to be constant over the element.

### §20.3.1. Stiffness Matrix

To obtain the plane beam-column stiffness in the local system we simply add the stiffness matrices derived in Chapters 12 and 13, respectively, to get

$$\bar{\mathbf{K}}^e = \frac{EA}{\ell}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 1 & 0 & 0 \\ & & & & 0 & 0 \\ \text{symm} & & & & & 0 \end{bmatrix} + \frac{EI}{\ell^3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 12 & 6\ell & 0 & -12 & 6\ell \\ & & 4\ell^2 & 0 & -6\ell & 2\ell^2 \\ & & & 0 & 0 & 0 \\ & & & & 12 & -6\ell \\ \text{symm} & & & & & 4\ell^2 \end{bmatrix} \tag{20.8}$$

```
PlaneBeamColumn2Stiffness[ncoor_,Em_,{A_,Izz_},options_]:=Module[
 {x1,x2,y1,y2,x21,y21,EA,EI,numer,L,LL,LLL,Te,Kebar,Ke},
  {{x1,y1},{x2,y2}}=ncoor; {x21,y21}={x2-x1,y2-y1};
   EA=Em*A; EI=Em*Izz; {numer}=options;
   LL=Simplify[x21^2+y21^2]; L=Sqrt[LL];
   If [numer,{x21,y21,EA,EI,LL,L}=N[{x21,y21,EA,EI,LL,L}]];
   If [!numer, L=PowerExpand[L]]; LLL=Simplify[LL*L];
   Kebar= (EA/L)*{
  { 1,0,0,-1,0,0},{0,0,0,0,0,0},{0,0,0,0,0,0},
  {-1,0,0, 1,0,0},{0,0,0,0,0,0},{0,0,0,0,0,0}} +
         (2*EI/LLL)*{
  { 0,0,0,0,0,0},{0, 6, 3*L,0,-6, 3*L},{0,3*L,2*LL,0,-3*L,  LL},
  { 0,0,0,0,0,0},{0,-6,-3*L,0, 6,-3*L},{0,3*L,  LL,0,-3*L,2*LL}};
   Te={{x21,y21,0,0,0,0}/L,{-y21,x21,0,0,0,0}/L,{0,0,1,0,0,0},
       {0,0,0,x21,y21,0}/L,{0,0,0,-y21,x21,0}/L,{0,0,0,0,0,1}};
   Ke=Transpose[Te].Kebar.Te;
   Return[Ke] ];
```

FIGURE 20.10. *Mathematica* module to form the stiffness matrix of a two-node, prismatic plane beam-column element.

The two matrices on the right of (20.8) come from the bar stiffness (12.22) and the Bernoulli-Euler bending stiffness (13.20), respectively. Before adding them, rows and columns have been rearranged in accordance with the nodal freedoms (20.7).

The displacement transformation matrix between local and global systems is

$$\bar{\mathbf{u}}^e = \begin{bmatrix} \bar{u}_{x1} \\ \bar{u}_{y1} \\ \theta_{z1} \\ u_{x2} \\ \bar{u}_{y2} \\ \theta_{z2} \end{bmatrix} = \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ \theta_{z1} \\ u_{x2} \\ u_{y2} \\ \theta_{z2} \end{bmatrix} = \mathbf{T}\,\mathbf{u}^e, \tag{20.9}$$

where $c = \cos\varphi = (x_2 - x_1)/\ell$, $s = \sin\varphi = (y_2 - y_1)/\ell$, and $\varphi$ is the angle between $\bar{x}$ and $x$, measured positive-counterclockwise from $x$; see Figure 20.9. The stiffness matrix in the global system is obtained through the congruential transformation

$$\mathbf{K}^e = \mathbf{T}^T\,\bar{\mathbf{K}}^e\,\mathbf{T}. \tag{20.10}$$

An explicit expression of the entries of $\mathbf{K}^e$ is messy, and (unlike the bar) it is better to let the program do it.

### §20.3.2. Stiffness Module

The computation of the stiffness matrix $\mathbf{K}^e$ of the two-node, prismatic plane beam-column element is done by *Mathematica* module `PlaneBeamColumn2Stiffness`. This is listed in Figure 20.10. The module is invoked as

```
Ke = PlaneBeamColumn2Stiffness[ncoor, Em, {A,Izz}, options]      (20.11)
```

The arguments are

```
ClearAll[L,Em,A,Izz];
ncoor={{0,0},{3,4}}; Em=100; A=125; Izz=250;
Ke= PlaneBeamColumn2Stiffness[ncoor,Em,{A,Izz},{True}];
Print["Numerical Elem Stiff Matrix: "];
Print[Ke//MatrixForm];
Print["Eigenvalues of Ke=",Chop[Eigenvalues[Ke]]];
```

Numerical Elem Stiff Matrix:

$$\begin{pmatrix} 2436. & 48. & -4800. & -2436. & -48. & -4800. \\ 48. & 2464. & 3600. & -48. & -2464. & 3600. \\ -4800. & 3600. & 20000. & 4800. & -3600. & 10000. \\ -2436. & -48. & 4800. & 2436. & 48. & 4800. \\ -48. & -2464. & -3600. & 48. & 2464. & -3600. \\ -4800. & 3600. & 10000. & 4800. & -3600. & 20000. \end{pmatrix}$$

Eigenvalues of Ke = {34800., 10000., 5000., 0, 0, 0}

FIGURE 20.11. Test of two-node plane beam-column element with numeric inputs.

| | |
|---|---|
| `ncoor` | Node coordinates of element arranged as `{ { x1,y1 },{ x2,y2 } }`. |
| `Em` | Elastic modulus. |
| `A` | Cross section area. |
| `Izz` | Moment of inertia of cross section area respect to axis $z$. |
| `options` | A list of processing options. For this element is has only one entry: `{ numer }`. This is a logical flag with the value `True` or `False`. If `True` the computations are carried out in floating-point arithmetic. If `False` symbolic processing is assumed. |

The module returns the $6 \times 6$ element stiffness matrix as function value.

### §20.3.3. Testing the Plane Beam-Column Module

The beam-column stiffness are tested by the scripts shown in Figures 20.11 and 20.12.

The script at the top of Figure 20.11 tests a numerically defined element of length $\ell = 5$ with end nodes located at (0, 0) and (3, 4), respectively, with $E = 100$, $A = 125$ and $I_{zz} = 250$. The output is shown at the bottom of that figure. The stiffness matrix returned in `Ke` is printed. Its six eigenvalues are computed and printed. As expected three eigenvalues, which correspond to the three independent rigid body motions of the element, are zero. The remaining three eigenvalues are positive.

The script at the top of Figure 20.12 tests a plane beam-column of length $L$ with end nodes at (0, 0) and $(3L/5, 4L/5)$. The properties $E$, $A$ and $I_{zz}$ are kept in symbolic form. The output is shown at the bottom of that figure. The printed matrix looks complicated because bar and beam coupling occurs when the element is not aligned with the global axes. The eigenvalues are obtained in closed symbolic form, and their simplicity provides a good check that the transformation matrix (20.9) is orthogonal. Three eigenvalues are exactly zero; one is associated with the axial (bar) stiffness and two with the flexural (beam) stiffness.

```
ClearAll[L,Em,A,Izz];
ncoor={{0,0},{3*L/5,4*L/5}};
Ke=  PlaneBeamColumn2Stiffness[ncoor,Em,{A,Izz},{False}];
Print["Symbolic Elem Stiff Matrix:"]; kfac=Em;
Ke=Simplify[Ke/kfac]; Print[kfac," ",Ke//MatrixForm];
Print["Eigenvalues of Ke=",kfac,"*",Eigenvalues[Ke]];
```

Symbolic Elem Stiff Matrix:

$$\mathrm{Em}\begin{pmatrix}
\frac{3\,(64\,\mathrm{Izz}+3\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & -\frac{3\,(64\,\mathrm{Izz}+3\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} \\
\frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{4\,(27\,\mathrm{Izz}+4\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & -\frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{4\,(27\,\mathrm{Izz}+4\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} \\
-\frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{4\,\mathrm{Izz}}{\mathrm{L}} & \frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & -\frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{2\,\mathrm{Izz}}{\mathrm{L}} \\
-\frac{3\,(64\,\mathrm{Izz}+3\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{3\,(64\,\mathrm{Izz}+3\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} \\
-\frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{4\,(27\,\mathrm{Izz}+4\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{12\,(-12\,\mathrm{Izz}+\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & \frac{4\,(27\,\mathrm{Izz}+4\,\mathrm{A\,L}^2)}{25\,\mathrm{L}^3} & -\frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} \\
-\frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{2\,\mathrm{Izz}}{\mathrm{L}} & \frac{24\,\mathrm{Izz}}{5\,\mathrm{L}^2} & -\frac{18\,\mathrm{Izz}}{5\,\mathrm{L}^2} & \frac{4\,\mathrm{Izz}}{\mathrm{L}}
\end{pmatrix}$$

Eigenvalues of Ke = Em∗$\{0, 0, 0, \frac{2\mathrm{A}}{\mathrm{L}}, \frac{2\,\mathrm{Izz}}{\mathrm{L}}, \frac{6\,(4\,\mathrm{Izz}+\mathrm{Izz\,L}^2)}{\mathrm{L}^3}\}$

FIGURE 20.12. Test of two-node plane beam-column element with symbolic inputs.

## §20.4. *The Space Beam Element

A second example in 3D is the general beam element shown in Figure 20.13. The element is prismatic and has two end nodes: 1 and 2, placed at the centroid of the end cross sections.

These define the local $\bar{x}$ axis as directed from 1 to 2. For simplicity the cross section will be assumed to be doubly symmetric, as is the case in commercial I and double-T profiles. The principal moments of inertia are defined by these symmetries. The local $\bar{y}$ and $\bar{z}$ axes are aligned with the symmetry lines of the cross section forming a RH system with $\bar{x}$. Consequently the principal moments of inertia are $I_{yy}$ and $I_{zz}$, the bars being omitted for convenience.

The global coordinate system is $\{x, y, z\}$. To define the orientation of $\{\bar{y}, \bar{z}\}$ with respect to the global system, a third orientation node 3, which must not be colinear with 1–2, is introduced. See Figure 20.13. Axis $\bar{y}$ lies in the 1–2–3 plane and $\bar{z}$ is normal to 1–2–3.

Six global DOF are defined at each node $i$: the 3 translations $u_{xi}$, $u_{yi}$, $u_{zi}$ and the 3 rotations $\theta_{xi}$, $\theta_{yi}$, $\theta_{zi}$.

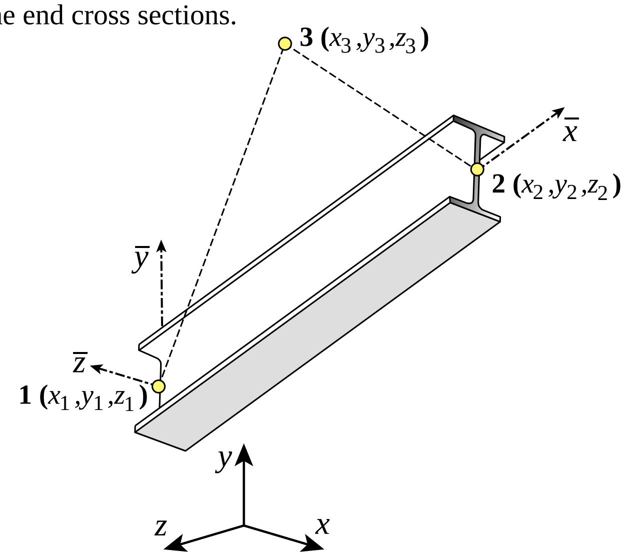


FIGURE 20.13. The space (3D) beam element.

### §20.4.1. *Stiffness Matrix

The element global node displacements and conjugate forces are arranged as

$$\begin{aligned}
\mathbf{u}^e &= [\, u_{x1} \;\; u_{y1} \;\; u_{z1} \;\; \theta_{x1} \;\; \theta_{y1} \;\; \theta_{z1} \;\; u_{x2} \;\; u_{y2} \;\; u_{z2} \;\; \theta_{x2} \;\; \theta_{y2} \;\; \theta_{z2} \,]^T, \\
\mathbf{f}^e &= [\, f_{x1} \;\; f_{y1} \;\; f_{z1} \;\; m_{x1} \;\; m_{y1} \;\; m_{z1} \;\; f_{x2} \;\; f_{y2} \;\; f_{z2} \;\; m_{x2} \;\; m_{y2} \;\; m_{z2} \,]^T.
\end{aligned} \tag{20.12}$$

The beam material is characterized by the elastic modulus $E$ and the shear modulus $G$ (the latter appears in the torsional stiffness). Four cross section properties are needed: the cross section area $A$, the moment of inertia

$J$ that characterizes torsional rigidity,[2] and the two principal moments of inertia $I_{yy}$ and $I_{zz}$ taken with respect to $\bar{y}$ and $\bar{z}$, respectively. The length of the element is denoted by $L$. The Bernoulli-Euler model is used; thus the effect of tranverse shear on the beam stiffness is neglected.

To simplify the following expressions, define the following "rigidity" combinations by symbols: $R^a = EA/L$, $R^t = GJ/L$, $R^b_{y3} = EI_{yy}/L^3$, $R^b_{y2} = EI_{yy}/L^2$, $R^b_y = EI_{yy}/L$, $R^b_{z3} = EI_{zz}/L^3$, $R^b_{z2} = EI_{zz}/L^2$, $R^b_z = EI_{zz}/L$. Note that $R^a$ is the axial rigidity, $R^t$ the torsional rigidity, while the $R^b$'s are bending rigities scaled by the length in various ways. Then the $12 \times 12$ local stiffness matrix can be written as[3]

$$\bar{\mathbf{K}}^e = \begin{bmatrix}
R^a & 0 & 0 & 0 & 0 & 0 & -R^a & 0 & 0 & 0 & 0 & 0 \\
0 & 12R^b_{z3} & 0 & 0 & 0 & 6R^b_{z2} & 0 & -12R^b_{z3} & 0 & 0 & 0 & 6R^b_{z2} \\
0 & 0 & 12R^b_{y3} & 0 & -6R^b_{y2} & 0 & 0 & 0 & -12R^b_{y3} & 0 & -6R^b_{y2} & 0 \\
0 & 0 & 0 & R^t & 0 & 0 & 0 & 0 & 0 & -R^t & 0 & 0 \\
0 & 0 & -6R^b_{y2} & 0 & 4R^b_y & 0 & 0 & 0 & 6R^b_{y2} & 0 & 2R^b_y & 0 \\
0 & 6R^b_{z2} & 0 & 0 & 0 & 4R^b_z & 0 & -6R^b_{z2} & 0 & 0 & 0 & 2R^b_z \\
-R^a & 0 & 0 & 0 & 0 & 0 & R^a & 0 & 0 & 0 & 0 & 0 \\
0 & -12R^b_{z3} & 0 & 0 & 0 & -6R^b_{z2} & 0 & 12R^b_{z3} & 0 & 0 & 0 & -6R^b_{z2} \\
0 & 0 & -12R^b_{y3} & 0 & 6R^b_{y2} & 0 & 0 & 0 & 12R^b_{y3} & 0 & 6R^b_{y2} & 0 \\
0 & 0 & 0 & -R^t & 0 & 0 & 0 & 0 & 0 & R^t & 0 & 0 \\
0 & 0 & -6R^b_{y2} & 0 & 2R^b_y & 0 & 0 & 0 & 6R^b_{y2} & 0 & 4R^b_y & 0 \\
0 & 6R^b_{z2} & 0 & 0 & 0 & 2R^b_z & 0 & -6R^b_{z2} & 0 & 0 & 0 & 4R^b_z
\end{bmatrix} \tag{20.13}$$

The transformation to the global system is the subject of Exercise 20.8.

### §20.4.2. *Stiffness Module

The computation of the stiffness matrix $\mathbf{K}^e$ of the two-node, prismatic plane beam-column element is done by *Mathematica* module `SpaceBeamColumn2Stiffness`. This is listed in Figure 20.14. The module is invoked as

```
Ke = SpaceBeamColumn2Stiffness[ncoor, {Em,Gm}, {A,Izz,Iyy,Jxx}, options]
```
(20.14)

The arguments are

| | |
|---|---|
| `ncoor` | Node coordinates of element arranged as `{{x1,y1,z1},{x2,y2,z2},{x3,y3,z3}}`, in which `{x3,y3,z3}` specifies an orientation node 3 that defines the local frame. See §20.4.1. |
| `Em` | Elastic modulus. |
| `Gm` | Shear modulus. |
| `A` | Cross section area. |
| `Izz` | Moment of inertia of cross section area respect to axis $\bar{z}$. |
| `Iyy` | Moment of inertia of cross section area respect to axis $\bar{y}$. |
| `Jxx` | Inertia with respect to $\bar{x}$ that appears in torsional rigidity $GJ$. |
| `options` | A list of processing options. For this element is has only one entry: `{numer}`. This is a logical flag with the value `True` or `False`. If `True` the computations are carried out in floating-point arithmetic. If `False` symbolic processing is assumed. |

The module returns the $12 \times 12$ element stiffness matrix as function value.

The implementation logic and testing of this element is the subject of Exercises 20.8 and 20.9.

---

[2] For circular and annular cross sections, $J$ is the polar moment of inertia of the cross section wrt $\bar{x}$. For other sections $J$ has dimensions of (length)$^4$ but must be calculated according to St. Venant's theory of torsion, or approximate theories.

[3] Cf. page 79 of Pzremieniecki [122]. The presentation in this book includes transverse shear effects as per Timoshenko's beam theory. The form (20.13) results from neglecting those effects.

```
SpaceBeamColumn2Stiffness[ncoor_,{Em_,Gm_},{A_,Izz_,Iyy_,Jxx_},
  options_]:= Module[
 {x1,x2,y1,y2,z1,z2,x21,y21,z21,xm,ym,zm,x0,y0,z0,dx,dy,dz,
  EA,EIyy,EIzz,GJ,numer,ra,ry,ry2,ry3,rz,rz2,rz3,rx,
  L,LL,LLL,yL,txx,txy,txz,tyx,tyy,tyz,tzx,tzy,tzz,T,Kebar,Ke},
  {x1,y1,z1}=ncoor[[1]]; {x2,y2,z2}=ncoor[[2]];
  {x0,y0,z0}={xm,ym,zm}={x1+x2,y1+y2,z1+z2}/2;
  If [Length[ncoor]<=2,{x0,y0,z0}+={0,1,0}];
  If [Length[ncoor]==3,{x0,y0,z0}=ncoor[[3]] ];
  {x21,y21,z21}={x2-x1,y2-y1,z2-z1}; {numer}=options;
  EA=Em*A; EIzz=Em*Izz; EIyy=Em*Iyy; GJ=Gm*Jxx;
  LL=Simplify[x21^2+y21^2+z21^2]; L=Sqrt[LL];
  If [numer, {x21,y21,z21,EA,EIyy,EIzz,GJ,LL,L}=
           N[{x21,y21,z21,EA,EIyy,EIzz,GJ,LL,L}]];
  If [!numer, L=PowerExpand[L]]; LLL=Simplify[LL*L];
  ra=EA/L; rx=GJ/L;
  ry=2*EIyy/L; ry2=6*EIyy/LL; ry3=12*EIyy/LLL;
  rz=2*EIzz/L; rz2=6*EIzz/LL; rz3=12*EIzz/LLL;
  Kebar={
  { ra,   0,   0,  0,   0,   0, -ra,   0,   0,   0,   0,   0},
  {  0, rz3,   0,  0,   0, rz2,   0,-rz3,   0,   0,   0, rz2},
  {  0,   0, ry3,  0,-ry2,   0,   0,   0,-ry3,   0,-ry2,   0},
  {  0,   0,   0, rx,   0,   0,   0,   0,   0, -rx,   0,   0},
  {  0,   0,-ry2,  0,2*ry,   0,   0,   0, ry2,   0,  ry,   0},
  {  0, rz2,   0,  0,   0,2*rz,   0,-rz2,   0,   0,   0,  rz},
  {-ra,   0,   0,  0,   0,   0,  ra,   0,   0,   0,   0,   0},
  {  0,-rz3,   0,  0,   0,-rz2,   0, rz3,   0,   0,   0,-rz2},
  {  0,   0,-ry3,  0, ry2,   0,   0,   0, ry3,   0, ry2,   0},
  {  0,   0,   0,-rx,   0,   0,   0,   0,   0,  rx,   0,   0},
  {  0,   0,-ry2,  0,  ry,   0,   0,   0, ry2,   0,2*ry,   0},
  {  0, rz2,   0,  0,   0,  rz,   0,-rz2,   0,   0,   0,2*rz}};
  {dx,dy,dz}={x0-xm,y0-ym,z0-zm};If[numer,{dx,dy,dz}=N[{dx,dy,dz}]];
  tzx=dz*y21-dy*z21; tzy=dx*z21-dz*x21; tzz=dy*x21-dx*y21;
  zL=Sqrt[tzx^2+tzy^2+tzz^2];
  If [!numer,zL=Simplify[PowerExpand[zL]]];
  {tzx,tzy,tzz}={tzx,tzy,tzz}/zL;   {txx,txy,txz}={x21,y21,z21}/L;
  tyx=tzy*txz-tzz*txy; tyy=tzz*txx-tzx*txz; tyz=tzx*txy-tzy*txx;
  Te={{txx,txy,txz,   0,  0,  0,   0,  0,  0,   0,  0,  0},
      {tyx,tyy,tyz,   0,  0,  0,   0,  0,  0,   0,  0,  0},
      {tzx,tzy,tzz,   0,  0,  0,   0,  0,  0,   0,  0,  0},
      {  0,  0,  0, txx,txy,txz,   0,  0,  0,   0,  0,  0},
      {  0,  0,  0, tyx,tyy,tyz,   0,  0,  0,   0,  0,  0},
      {  0,  0,  0, tzx,tzy,tzz,   0,  0,  0,   0,  0,  0},
      {  0,  0,  0,   0,  0,  0, txx,txy,txz,   0,  0,  0},
      {  0,  0,  0,   0,  0,  0, tyx,tyy,tyz,   0,  0,  0},
      {  0,  0,  0,   0,  0,  0, tzx,tzy,tzz,   0,  0,  0},
      {  0,  0,  0,   0,  0,  0,   0,  0,  0, txx,txy,txz},
      {  0,  0,  0,   0,  0,  0,   0,  0,  0, tyx,tyy,tyz},
      {  0,  0,  0,   0,  0,  0,   0,  0,  0, tzx,tzy,tzz}};
    Ke=Transpose[Te].Kebar.Te;
    Return[Ke]
  ];
```

FIGURE 20.14. Module to form stiffness of space (3D) beam.

## Notes and Bibliography

All elements implemented here are formulated in most books dealing with matrix structural analysis. Przemieniecki [122] has been recommended in Chapter 1 on account of being inexpensive. The implementation and testing procedures are not covered in any book.

## Homework Exercises for Chapter 20
## Implementation of One-Dimensional Elements

**EXERCISE 20.1** [C:15] Download the plane bar stiffness module and their testers and verify the test results reported here. Comment on whether the stiffness matrix `Ke` has the correct rank of 1.

**EXERCISE 20.2** [C:15] Download the space bar stiffness module and their testers and verify the test results reported here. Comment on whether the computed stiffness matrix `Ke` has the correct rank of 1.

**EXERCISE 20.3** [C:15] Download the plane beam-column stiffness module and their testers and verify the test results reported here. Comment on whether the computed stiffness matrix `Ke` has the correct rank of 3.

**EXERCISE 20.4** [A+C:30] Explain why the space bar element has rank 1 although it has 6 degrees of freedom and 6 rigid body modes. (According to the formula given in Chapter 19, the correct rank should be $6 - 6 = 0$.)

**EXERCISE 20.5** [C:25] Implement the plane bar, plane beam-column and space bar stiffness element module in a lower level programming language and check them by writing a short test driver. [Do not bother about the mass modules.] Your choices are C, Fortran 77 or Fortran 90. (C++ is overkill for this kind of software).

**EXERCISE 20.6** [A:25] Explain why the eigenvalues of $\mathbf{K}^e$ of any the elements given here do not change if the $\{x, y, z\}$ global axes change.

**EXERCISE 20.7** [A+C:30] (Advanced) Implement a 3-node space bar element. *Hint:* use the results of Exercise 16.5 and transform the local stiffness to global coordinates via a $3 \times 9$ transformation matrix. Test the element and verify that it has two nonzero eigenvalues.

**EXERCISE 20.8** [D+A:25] Explain the logic of the space beam module listed in Figure 20.14. Assume that $\bar{\mathbf{K}}^e$ stored in `Kebar` is correct; focus instead on how the local to global transformation is built and applied.

**EXERCISE 20.9** [C:25] Test the space beam element of Figure 20.14 using the scripts given in Figures E20.1 and E20.2, and report results. Comment on whether the element has the correct rank of 6.

```
ClearAll[L,Em,Gm,A,Izz,Iyy,Jxx];
ncoor={{0,0,0},{1,8,4}}; Em=54; Gm=30;
A=18; Izz=36; Iyy=72; Jxx=27;
Ke= SpaceBeamColumn2Stiffness[ncoor,{Em,Gm},{A,Izz,Iyy,Jxx},{True}];
Print["Numerical Elem Stiff Matrix: "];
Print[SetPrecision[Ke,4]//MatrixForm];
Print["Eigenvalues of Ke=",Chop[Eigenvalues[Ke]]];
```

FIGURE E20.1. Script for numeric testing of the space beam module of Figure 20.14.

```
ClearAll[L,Em,Gm,A,Izz,Iyy,Jxx];
ncoor={{0,0,0},{2*L,2*L,L}/3};
Ke=SpaceBeamColumn2Stiffness[ncoor,{Em,Gm},{A,Izz,Iyy,Jxx},{False}];
kfac=Em; Ke=Simplify[Ke/kfac];
Print["Numerical Elem Stiff Matrix: "];
Print[kfac," ",Ke//MatrixForm];
Print["Eigenvalues of Ke=",kfac,"*",Eigenvalues[Ke]];
```

FIGURE E20.2. Script for symbolic testing of the space beam module of Figure 20.14.