Computer Implementation 4.3 *(Matlab)* *Beam element (p. 262)*

The analysis of beams can be performed conveniently by writing two small *Matlab* functions, one for defining the element stiffness matrix and the other for computing the element displacement, bending moment and shear force. For each element the results are computed at element ends and at center of the element. By adjusting the increment of s in the for loop in the BeamResults function the results can be obtained at several points and used for plotting bending moment and shear force diagrams.

MatlabFiles\Chap4\BeamElement.m

```
function [ke, rq] = BeamElement(EI, q, coord)
% [ke, rq] = BeamElement(EI, q, coord)
% Generates equations for a beam element
% EI = beam stiffness
% q = distributed load
% coord = coordinates at the element ends

L=coord(2)-coord(1);
ke = [(12*EI)/L^3, (6*EI)/L^2, -((12*EI)/L^3), (6*EI)/L^2;
   (6*EI)/L^2, (4*EI)/L, -((6*EI)/L^2), (2*EI)/L;
   -((12*EI)/L^3), -((6*EI)/L^2), (12*EI)/L^3, -((6*EI)/L^2);
   (6*EI)/L^2, (2*EI)/L, -((6*EI)/L^2), (4*EI)/L];
rq = [(L*q)/2; (L^2*q)/12; (L*q)/2; -((L^2*q)/12)];
```

MatlabFiles\Chap4\BeamResults.m

```
function [v, bm, V] = BeamResults(EI, q, coord, dn)
% [v, bm, V] = BeamResults(EI, q, coord, dn)
% Generates beam element results
% EI = beam stiffness
% q = distributed load
% coord = coordinates at the element ends
% dn = nodal solution
L=coord(2)-coord(1);
v=[]; bm=[]; V=[];
% Change increment to get results at more points
for s=0:L/2:L
   n = [(2*s^3)/L^3 - (3*s^2)/L^2 + 1, s^3/L^2 - (2*s^2)/L + s, ...
         (3*s^2)/L^2 - (2*s^3)/L^3, s^3/L^2 - s^2/L];
   v = [v; [coord(1)+s, n*dn+(q*(L - s)^2*s^2)/(24*EI)]];
   dn2 = [(12*s)/L^3 - 6/L^2, (6*s)/L^2 - 4/L, 6/L^2 - (12*s)/L^3, ...
         (6*s)/L^2 - 2/L];
   bm = [bm; [coord(1)+s, EI*dn2*dn+(q*(L^2 - 6*s*L + 6*s^2))/(12)]];
   dn3 = [12/L^3, 6/L^2, -(12/L^3), 6/L^2];
```

```
    V = [V; [coord(1)+s, EI*dn3*dn+((q*(12*s - 6*L))/(12))]];
end
```

Using these functions now we consider solution of the three element model. The steps are exactly those used in other *Matlab* implementations.

MatlabFiles\Chap4\BeamEx.m

```
% Beam example
F = 18; q = 10; EI = (210*10^6)*4*10^(-4);
L=2; nodes =[0:L:3*L]; n=2*length(nodes);
debc=[1,2,7]; ebcVals=zeros(length(debc),1);
K=zeros(n); R = zeros(n,1); R(3)=-F;
% Generate equations for each element and assemble them.
for i=1:2
   lm=[2*(i-1)+1,2*(i-1)+2,2*(i-1)+3,2*(i-1)+4];
   [ke, rq] = BeamElement(2*EI, 0, nodes([i:i+1]));
   K(lm, lm) = K(lm, lm) + ke;
   R(lm) = R(lm) + rq;
end
for i=3
   lm=[2*(i-1)+1,2*(i-1)+2,2*(i-1)+3,2*(i-1)+4];
   [ke, rq] = BeamElement(EI, -q, nodes([i:i+1]));
   K(lm, lm) = K(lm, lm) + ke;
   R(lm) = R(lm) + rq;
end
K
R
% Nodal solution and reactions
d = NodalSoln(K, R, debc, ebcVals)
va=[]; bma=[]; Va=[];
for i=1:2
   lm=[2*(i-1)+1,2*(i-1)+2,2*(i-1)+3,2*(i-1)+4];
   [v, bm, V]=BeamResults(2*EI, 0, nodes([i:i+1]), d(lm));
   va = [va; v]; bma = [bma; bm]; Va = [Va; V];
end
for i=3
   lm=[2*(i-1)+1,2*(i-1)+2,2*(i-1)+3,2*(i-1)+4];
   [v, bm, V]=BeamResults(EI, -q, nodes([i:i+1]), d(lm));
   va = [va; v]; bma = [bma; bm]; Va = [Va; V];
end
va
bma
Va
```

```
>> BeamEx
```

K =

Columns 1 through 6

```
   252000   252000  -252000   252000        0        0
   252000   336000  -252000   168000        0        0
  -252000  -252000   504000        0  -252000   252000
   252000   168000        0   672000  -252000   168000
        0        0  -252000  -252000   378000  -126000
        0        0   252000   168000  -126000   504000
        0        0        0        0  -126000  -126000
        0        0        0        0   126000    84000
```

Columns 7 through 8

```
        0        0
        0        0
        0        0
        0        0
  -126000   126000
  -126000    84000
   126000  -126000
  -126000   168000
```

R =

```
        0
        0
      -18
        0
      -10
  -3.3333
      -10
   3.3333
```

d =

```
        0
        0
 -0.00021315
 -0.00013138
 -0.00034127
 1.3605e-005
        0
```

0.00026899

va =

0	0
1	-7.3732e-005
2	-0.00021315
2	-0.00021315
3	-0.00031346
4	-0.00034127
4	-0.00034127
5	-0.00023944
6	0

bma =

0	-31.643
1	-11.036
2	9.5714
2	9.5714
3	12.179
4	14.786
4	14.786
5	12.393
6	-1.199e-014

Va =

0	20.607
1	20.607
2	20.607
2	2.6071
3	2.6071
4	2.6071
4	2.6071
5	-7.3929
6	-17.393