A Finite Element Solution of the Beam Equation via MATLAB

S Rao. Gunakala
Department of Mathematics and Statistics
The University of the West Indies
St. Augustine. Trinidad and Tobago

D.M.G. Comissiong
Department of Mathematics and Statistics
The University of the West Indies
St. Augustine. Trinidad and Tobago

K.Jordan
Department of Mathematics and Statistics
The University of the West Indies
St. Augustine. Trinidad and Tobago

Alana Sankar
Department of Mathematics and Statistics
The University of the West Indies
St. Augustine. Trinidad and Tobago

Abstract

The vertical deflection of a simply supported and clamped beam is considered under a uniform load using the finite element method. The problem is solved using homogenous and non-homogenous boundary conditions with various numbers of elements. The governing differential equation is that pre-described by the Bernoulli beam which is a fourth order differential equation. Cubic elements are used as required for continuity. Graphs are presented and discussed for different loads in each case.

Keywords: Finite element method, Beam equation, Homogenous and non-homogenous conditions

1. Introduction

Beam equations have historical importance, as they have been the focus of attention for prominent scientists such as Leonardo da Vinci (14th Century) and Daniel Bernoulli (18thCentury). Practical applications of the beam equations are evident in mechanical structures built under the premise of beam theory. The importance of beam theory has been outlined in the literature over the years(see for example [3], [4], [5]). Examples include the construction of high-rise buildings, bridges across the rivers, air craft and heavy motor vehicles. In these structures, beams are used as the basis of supporting structures or as the main-frame foundation inaxles. Without a proper knowledge of beam theory, the successful manufacture of such structures would be unfeasible and unsafe. The Euler-Bernoulli beam theory, sometimes called the classical beam theory, is the most commonly used. It is simple and provides reasonable engineering approximations for many problems.In the paper, we shall illustrate the use of the Galerkin Finite Element Method to solve the beam equation with aid of Matlab.

The Finite Element Method (FEM) is one of the most powerful tools used in structural analysis. Finite Element Analysis is based on the premise that an approximate solution to any complex engineering problem can be reached by subdividing a larger complex structure into smaller non-overlapping components of simple geometry called finite elementsor elements.Complex partial differential equations that describe these structures can be reduced to a set of linear equations that can easily be solved using this method.

Elements are interconnected by points called nodes. Nodes will have nodal (vector) displacements or degrees of freedom which may include translations, rotations, and for special applications, higher order derivatives of displacements. In the Galerkin finite element method a trial function is substituted into the governing equations and the unknown node values are determined.

2. Governing Equation

In the Euler-Bernoulli beam theory, the transverse deflection u of the beam is governed by the fourth order differential equation

$$\frac{d^2}{dx^2}\left[r(x)\frac{d^2u}{dx^2}\right] = f(x,u), 0 \le x \le L \quad (1)$$

subject to the free end boundary conditions

$$u(0) = a_0, \quad \frac{d^2u(0)}{dx^2} = b_0,$$
$$u(L) = a_L, \quad \frac{d^2u(L)}{dx^2} = b_L. \quad (2)$$

The function $r(x) = E * I$ is the product of Young's modulus of elasticity E and moment of inertia I of the beam. It is referred to as the flexural rigidity, and is a measure of strength. The transversely distributed load is $f(x,u)$, while $u(x)$ is the transverse deflection the beam. In the linear case, $f(x,u) = q(x)u + p(x)$, and the beam equation (1) becomes

$$\frac{d^2}{dx^2}\left[EI\frac{d^2u}{dx^2}\right] = q(x)u + p(x), \ \ 0 \le x \le L, \quad (3)$$

where $q(x)$ is the coefficient of ground elasticity and $p(x)$ is a load force normal to the beam at the point x. For the linear case $f(x,u) = q(x)u + p(x)$, where $u(x)$ is the deflection of the beam, $q(x)$is the coefficient of ground elasticity, and $p(x)$ is the uniform load applied normal to the beam.

When the beam is supported by free ends and$q(x) = 0$, the solution $u(x)$describes the deflection of the beam under the load $p(x)$. In this case, the governing equations are

$$\frac{d^2}{dx^2}\left[EI\frac{d^2u}{dx^2}\right] = p(x) \ , \ 0 \le x \le L, \quad (4)$$
$$u(0) = a_0, \quad \frac{d^2u(0)}{dx^2} = b_0,$$
$$u(L) = a_L, \quad \frac{d^2u(L)}{dx^2} = b_L. \quad (5)$$

For simple data functions $f(x,u)$and $r(x)$, the exact solution of beam equation with boundary condition can be found by standard methods that are well known in literature of ordinary differential equations and their applications . For more developed data functions, when exact methods fail, numerical methods can be successfully applied to find an approximate solution for a broad class of boundary value problems. In the next section, we shall utilize the Galerkin Finite Element Method (FEM) to solve the boundary value problem (4)-(5) when the flexural rigidity function $r(x) = E * I$ is constant. In this case, equation (4) can be written as

$$EI\frac{d^4u}{dx^4} = p(x) \ , \ 0 \le x \le L. \quad (6)$$

3. Galerkin Finite Element Method

The first step in the Galerkin FEM is the discretization of the domain. Here, the domain of the problem (length of the beam) is divided into a finite set of line elements, each of which has at least two end nodes. Geometrically the element is the same as that used for bars. The second step is to obtain the weak form of the differential equation. For this purpose, we multiply the residual of differential equation (4) by a weight function $w(x)$ and integrate it by parts so as to evenly distribute the order of differentiation on u and w. The result is the equation

$$\int_0^L \left[EI\frac{d^4u}{dx^4} - p(x)\right] w \, dx = EI\frac{d^3u}{dx^3}w\Big]_0^L - EI\frac{d^2u}{dx^2}\frac{dw}{dx}\Big]_0^L + \int_0^L \left[EI\frac{d^2w}{dx^2}\frac{d^2u}{dx^2} - pw \, dx\right] = 0. \quad (7)$$

After obtaining the weak form, we proceed to choose the suitable approximating functions for the elements. It can be seen that the highest order of the derivative on $u(x)$ in the weak form (7) is three; therefore we choose an approximating function that is thrice differentiable. A cubic interpolation polynomial satisfies this requirement [1].Using the Galerkin FEM, we equate the weight function to the approximating function, $w_i = N_i$ where these cubic interpolation functions are known as Hermite cubic interpolation (or cubic spline) functions which are defined as

$$N_1 = 1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3, N_2 = x\left(1-\frac{x}{h}\right)^2, N_3 = 3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3, N_4 = x\left[\left(\frac{x}{h}\right)^2 - \frac{x}{h}\right].$$

On substituting these shape functions into the weak form of the equation (7) and assuming $u = \sum_{j=1}^{4} u_j N_j$, we get

$$\int_0^L \left[EI\frac{d^4u}{dx^4} - p(x)\right] w\, dx = EIu_{,xxx} N_i]_0^L - EIu_{,xx} N_{i,x}]_0^L + \int_0^L EI\, N_{i,xx} u_{,xx}\, dx - \int_0^L pN_i\, dx = 0.$$

The stiffness matrix is given by

$$K_{ij} = \int_0^L \frac{d^2N_i}{dx^2}\frac{d^2N_j}{dx^2}\ dx.$$

The force vector is

$$f_i = \int_0^L pN_i\, dx.$$

For the first element

$$K_{11} = \int_0^L \frac{d^2N_1}{dx^2}\frac{d^2N_1}{dx^2}\ dx = \int_0^L \frac{1}{L^3}(12x-6L)\frac{1}{L^3}(12x-6L)\, dx$$
$$= \frac{1}{L^6}\int_0^L 144x^2 - 144xL + 36L^2\ dx$$
$$= \frac{1}{L^6}[48x^3 - 72xL + 36xL]_0^L \qquad = \frac{1}{L^6}[12L^3] = \frac{12}{L^3}.$$

The remaining elements are found in a similar manner. The stiffness matrix becomes

$$\frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}.$$

Similarly, we can obtain the force vector matrix. The first value in the force vector is evaluated below.

$$\int_0^L p\left(1 - \frac{3x^2}{h^2} + \frac{2x^3}{h^3}\right)\, dx = p\left[x - \frac{x^3}{h^2} + \frac{x^4}{2h^3}\right]_0^L = p\left(L - \frac{L^3}{L^2} + \frac{L^4}{2L^3}\right) = p\left(\frac{L}{2}\right).$$

The remaining values are obtained in a similar manner using their corresponding shape functions. The resulting force vector is given as

$$f = \frac{Lp}{2}\begin{bmatrix} 1 \\ 6L \\ 1 \\ -6L \end{bmatrix}.$$

The corresponding system can represented as

$$\frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \frac{Lp}{2}\begin{bmatrix} 1 \\ 6L \\ 1 \\ -6L \end{bmatrix}$$

[Stiffness matrix][Displacement matrix] = force vector.

The system of equations is solved using MATLAB. The displacement for each element is solved under different conditions prescribed. Results were found for various numbers of elements under different loads.

4. Non-homogenous case

We consider the beam equation

$$EI\frac{d^4u}{dx^4} = p(\pi^4 x\sin\pi x - 4\pi^3\cos\pi x), \qquad 0 \le x \le L, \qquad (5a)$$

with corresponding non-homogenous boundary conditions

$$u(0) = 0, \qquad u''(0) = 2\pi$$
$$u(L) = 0, \qquad u''(L) = -2\pi.$$

It must be noted that $u(x) = x\ sin\ \pi x$ is the exact solution of the boundary value problem. In order to transform the non-homogenous boundary conditions into homogenous conditions, we introduce a new unknown

$$u(x) = u^0(x) + w(x), \qquad 0 \le x \le L,$$

where the interpolating polynomial w(x) is obtained by [6]:

$$w(x) = -\pi(x-1)^2 x^2 (2x-1).$$

Our problem becomes

$$EI\frac{d^4u^0}{dx^4} = p(\pi^4 x sin\pi x - 4\pi^3 cos\pi x + 240\pi x - 120\pi), \qquad 0 \le x \le L,$$

$$u^0(0) = 0, \qquad \frac{d^2u^0(0)}{dx^4} = 0$$

$$u^0(L) = 0, \qquad \frac{d^2u^0(L)}{dx^4} = 0.$$

The left hand side of the equation yields the same stiffness matrix as previously obtained. The right hand side can be represented as

$$\int_0^L p(\pi^4 x sin\pi x - 4\pi^3 cos\pi x + 240\pi x - 120\pi)N_i\ dx.$$

We obtain for N_1

$$\begin{aligned}
&p\int_0^L (\pi^4 x sin\pi x - 4\pi^3 cos\pi x + 240\pi x - 120\pi)N_1\ dx \\
&= p\int_0^L (\pi^4 x sin\pi x - 4\pi^3 cos\pi x + 240\pi x - 120\pi)\left(1 - \tfrac{3x^2}{h^2} + \tfrac{2x^3}{h^3}\right)\ dx \\
&+ p\int_0^L \frac{-3\pi^4}{h^2}x^3 sin\pi x + \frac{12\pi^3}{h^2}x^2 cos\pi x - \frac{720\pi}{h^2}x^3 + \frac{360\pi}{h^2}x^2\ \ dx \\
&+ p\int_0^L \frac{2\pi^4}{h^3}x^4 sin\pi x - \frac{8\pi^3}{h^3}x^3 cos\pi x + \frac{480\pi}{h^3}x^4 - \frac{240\pi}{h^3}x^3\ \ dx \\
&= \left(25\pi^2 - 9\pi + \frac{18}{\pi L^2} - \frac{48}{L^2}\right)sin\pi L + \left(3\pi^2 L - 3\pi^3 L - \frac{18}{L} + \frac{288\pi}{L}\right)cos\pi L - 156\pi L^2 - 60\pi L
\end{aligned}$$

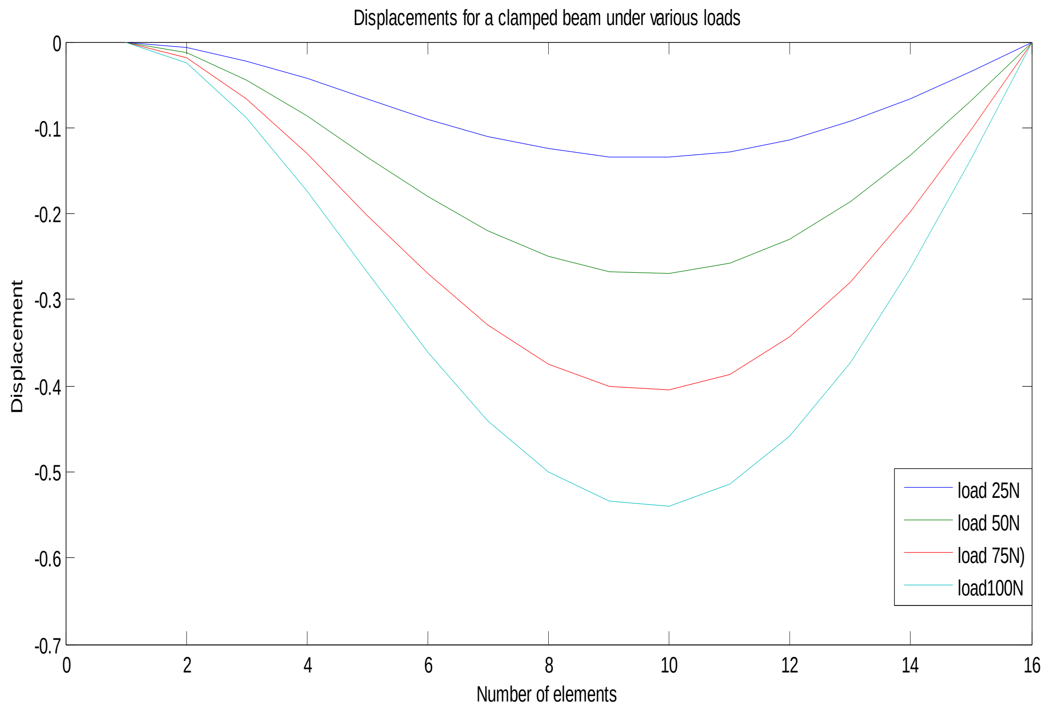
The remaining values are obtained in a similar manner using their corresponding shape functions. The resulting force vector is given as

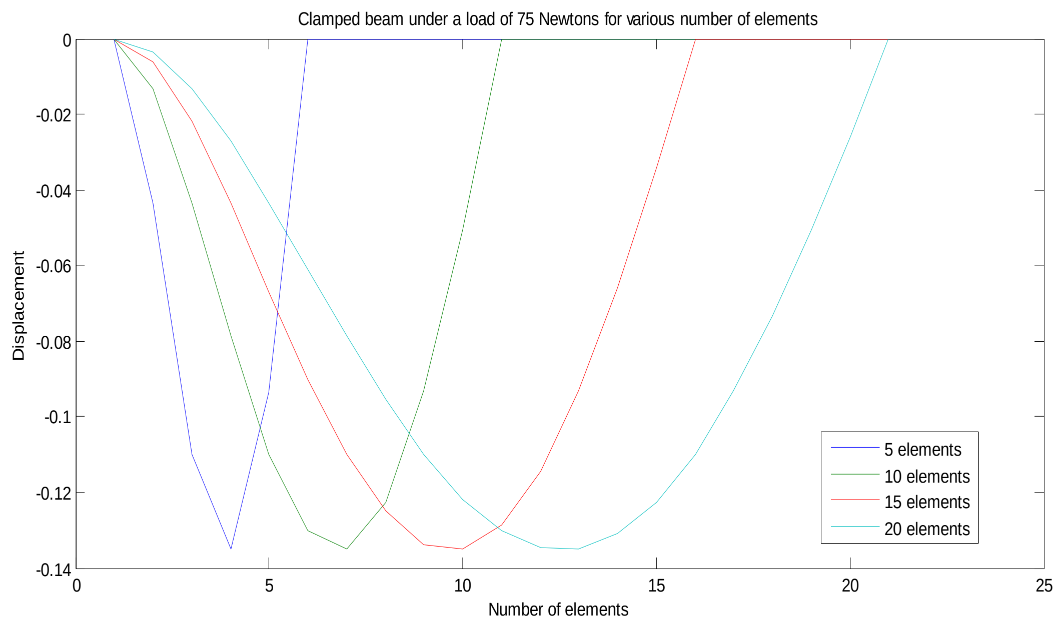
$$f = p\begin{bmatrix}
\left(25\pi^2 - 9\pi + \frac{18}{\pi L^2} - \frac{48}{L^2}\right)sin\pi L + \left(3\pi^2 L - 3\pi^3 L - \frac{18}{L} + \frac{288\pi}{L}\right)cos\pi L - 156\pi L^2 - 60\pi L \\
\frac{44}{L}sin\pi L + \left(2\pi + \frac{48}{\pi L^2}\right)cos\pi L - 98\pi L^3 - 10\pi L^2 \\
\left(\frac{6}{L^2} - 3\pi^2\right)sin\pi L - \left(\pi^3 L + \frac{6\pi}{L} + \frac{48}{\pi L^2} + \frac{48}{\pi L^3}\right)cos\pi L + 84\pi L^2 - 60\pi L \\
(48L + \pi^2 L^2 - 2)sin\pi L + \left(\pi^3 L^3 - \pi^3 L^4 + 2\pi L + \frac{48}{\pi}\right)cos\pi L - 120\pi L^4 + 80\pi L^3
\end{bmatrix}$$

5. Results

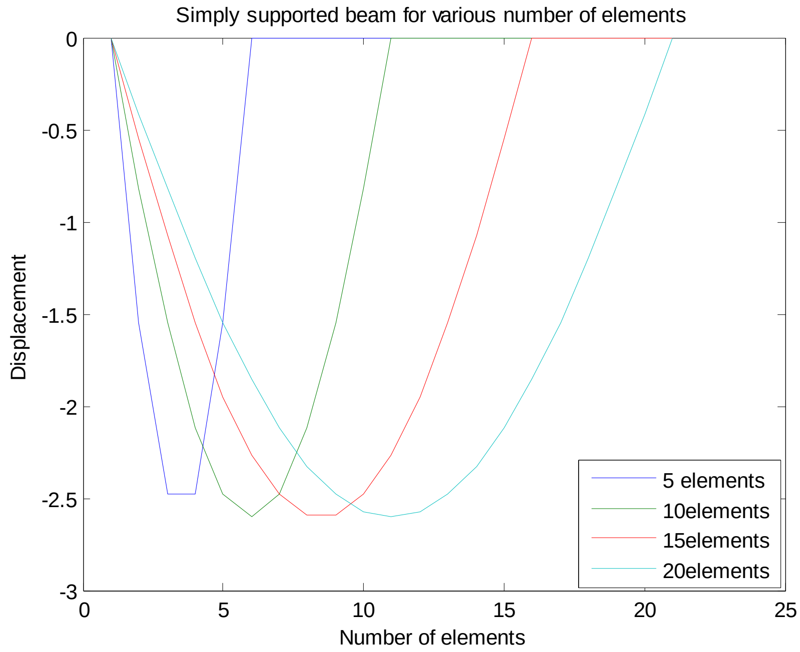
The results were first obtained for a beam clamped at both ends. The first graph represents a beam under the same load for various numbers of elements. The second graph displays the clamped beam under various loads. The code used to generate these results is given in Appendix A.

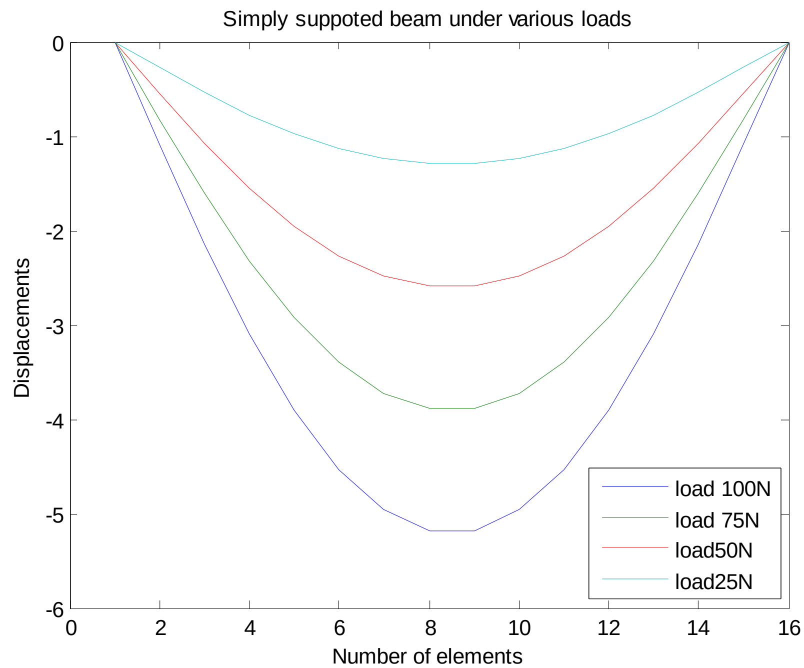
Graphs for the beam equation using homogenous boundary conditions



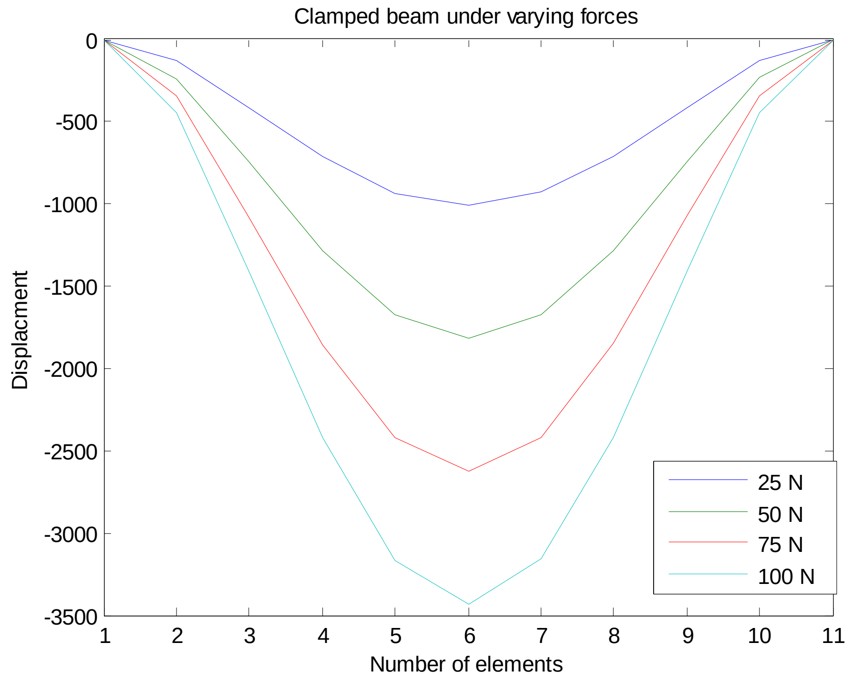


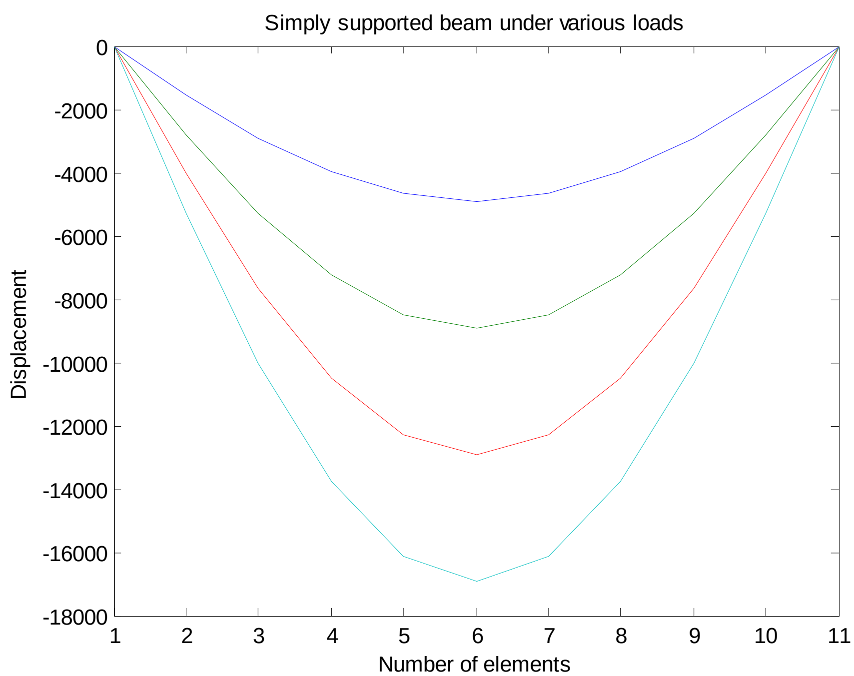
The boundary conditions were then changed to obtain results for a simply-supported beam. The graphical results were obtained for the same parameters as described for a clamped beam. The code used to obtain these results is given in Appendix B.





Graphs for the beam equation using non-homogenous boundary conditions





6. Discussion

The graphs display the results obtained for homogenous and non-homogenous boundary conditions. When the number of elements was increased for both clamped and simply-supported beams, the graphs yielded greater continuity which resulted from greater accuracy. That is to say the approximate solution was closer to the exact solution or there was less error as the number of elements increased. This is consistent with the theory of the FEM, as increasing the number of elements reduces the error, which in turn improves the accuracy of the solution. In addition, increasing the load also increases the displacement at each point on the beam. It should also be noted that the displacement for the simply-supported beam is greater than that of the clamped beam. Furthermore, the deflection for the non-homogenous boundary conditions was much more significant than that for the homogenous case.

8. References

A.J.M. Fereira (2009).*Matlab Codes for Finite Element Analysis, Solids and Structures.*

D.Logan (2007). *A First Course in the Finite Element Method (4th ed.).*

E.A. Witmer (1991-1992). "Elementary Bernoulli-Euler Beam Theory". *MIT Unified Engineering Course Notes*. pp. 5–114 to 5–164.

Euler-Bernoulli Beam Equation, http://en.wikipedia/Euler-Bernoulli-beam equation

E.P Popov., Mechanics of Materials, Prentice-Hall, 1978.

K.S. Thankane and T. Stys.(2009). Finite Difference Method for Beam Equation with Free Ends Using Mathematica. *Southern African Journal of Pure and Applied Mathematics*

McKenzie, William (2006). *Examples in Structural Analysis*. Taylor & Francis.

R.W Clough and J. Penzien .,Dynamics of Structures, McGraw-Hill, 1993.

Seon M. Han, Haym Benaroya and Timothy Wei (1999). *Journal of Sound and Vibration* (1999) 225(5), 935}988

Seon M. Han, Haym Benaroya and Timothy Wei (March 22, 1999) (PDF). *Dynamics of Transversely Vibrating Beams using four Engineering Theories*. final version. Academic Press. http://csxe.rutgers.edu/research/vibration/51.pdf. Retrieved 2007-04-15.

Y.Kwon and H.Bang(2000). *The Finite Element Method using Matlab (2nd ed.).*

6. Appendix

Appendix A: Finite Element Code for Clamped Beam

```
function [stiffness force displacements U reactions]= formstiffness(m,P)
 nodeCoordinates=linspace(0,1,m+1)';
xx=nodeCoordinates;
L=max(nodeCoordinates);
numberNodes=size(nodeCoordinates,1);
xx=nodeCoordinates(:,1);
E=1; I=1; EI=E*I;
GDof=2*numberNodes;
U=zeros(GDof,1);
force=zeros(GDof,1);
stiffness=zeros(GDof);
displacements=zeros(GDof,1);
for i=1:m;
   elementNodes(i,1)=i;
   elementNodes(i,2)=i+1;
end
 for e=1:m;
     indice=elementNodes(e,:);
   elementDof=[ 2*(indice(1)-1)+1  2*(indice(2)-1)  2*(indice(2)-1)+1  2*(indice(2)-1)+2];
   LElem=xx(indice(2))-xx(indice(1));
    f1=4*[P*LElem/2  P*LElem*LElem/12  P*LElem/2  -P*LElem*LElem/12]';
   force(elementDof)=force(elementDof)+f1;
   k1=EI/(LElem)^3*[12 6*LElem -12 6*LElem ;
            6*LElem 4*LElem^2  -6*LElem  2*LElem^2;
            -12 -6*LElem 12 -6*LElem;
           6*LElem 2*LElem^2 -6*LElem 4*LElem^2];

stiffness(elementDof,elementDof)=stiffness(elementDof,elementDof)+k1;
end
fixedNodeU=[1 2*m+1]';fixedNodeV=[2 2*m+1]';
prescribedDof=[fixedNodeU;fixedNodeV];
activeDof=setdiff([1:GDof]',[prescribedDof]);
 U=stiffness(activeDof,activeDof)\force(activeDof);
displacements=zeros(GDof,1);
displacements(activeDof)=U;
```

```
 disp('Displacements')
 plot(U)

jj=1:GDof; format
 [jj'  displacements];
F=stiffness*displacements;
reactions=F(prescribedDof);
disp('reactions')
[prescribedDof  reactions];

U=displacements(1:2:2*numberNodes);
plot(nodeCoordinates,U,'.')
```

Appendix B: Finite Element Code for SimplySupported Beam

```
function [stiffness force displacements U reactions]= formstiffness(m,P)

nodeCoordinates=linspace(0,1,m+1)';
xx=nodeCoordinates;
L=max(nodeCoordinates);
numberNodes=size(nodeCoordinates,1);
xx=nodeCoordinates(:,1);

 E=1; I=1; EI=E*I;
GDof=2*numberNodes;
U=zeros(GDof,1);
force=zeros(GDof,1);
stiffness=zeros(GDof);
displacements=zeros(GDof,1);

for i=1:m;
   elementNodes(i,1)=i;
   elementNodes(i,2)=i+1;
end

for e=1:m;

   indice=elementNodes(e,:);
   elementDof=[ 2*(indice(1)-1)+1  2*(indice(2)-1)  2*(indice(2)-1)+1  2*(indice(2)-1)+2];
   LElem=xx(indice(2))-xx(indice(1));

   f1=4*[P*LElem/2  P*LElem*LElem/12  P*LElem/2  -P*LElem*LElem/12]';

   force(elementDof)=force(elementDof)+f1;

   k1=EI/(LElem)^3*[12 6*LElem -12 6*LElem ;
            6*LElem 4*LElem^2  -6*LElem  2*LElem^2;
            -12 -6*LElem 12 -6*LElem;
           6*LElem 2*LElem^2 -6*LElem 4*LElem^2];

     stiffness(elementDof,elementDof)=stiffness(elementDof,elementDof)+k1;

end

fixedNodeU=[1 2*m+1]';fixedNodeV=[]';
prescribedDof=[fixedNodeU;fixedNodeV];
activeDof=setdiff([1:GDof]',[prescribedDof]);

U=stiffness(activeDof,activeDof)\force(activeDof);
displacements=zeros(GDof,1);
displacements(activeDof)=U;

disp('Displacements')

plot(U)
```