UNIVERSITY OF CALIFORNIA BERKELEY
Department of Civil Engineering
Spring 2011

Structural Engineering,
Mechanics and Materials
Professor: S. Govindjee

Mechanics of Structures (CE130N)
Labs 10-11

1 Objective

The objective of this lab is to program a virtual work based solution using a special set of Bubnov-Galerkin functions known as Hermite polynomials. These function underly the most important numerical method for solving beam problems – the finite element method. Unlike the polynomials we have been using, they produce matricies with reasonable properties as the number of approximation terms grows. To keep things simple will assume a linear elastic material but the ideas and methodology are quite general.

2 Model Problem

The model problem we will work with is a linear elastic beam which has an imposed displacement at the right-end, is built-in at the left-end and is subjected to a constant distributed load. Thus the problem we wish to solve looks like:

$$\frac{d^2M}{dx^2} = q \tag{1}$$

$$M = EI\kappa \tag{2}$$

$$\kappa = v'' \tag{3}$$

where $v(0) = 0$, $\theta(0) = 0$, $v(L) = 0.1$, $M(L) = EIv''(L) = 0$, $q(x) = -3000$, $EI = 120 \times 10^6$ with $L = 30$ – all in US customary units.

2.1 Exercise 1

Sketch the problem described above.

3 Function spaces

To deal with the boundary conditions we will employ the methodology we have used several times this semester of ignoring kinematic boundary conditions in the formal derivation of the matrix equations. We will then impose the boundary conditions at the end by separating the free degrees of freedom from the driven ones; see Lab 7. In this setting, this means

$$\mathcal{S} = \{v(x) \mid \text{no restrictions}\} \tag{4}$$

and

$$\mathcal{V} = \{\delta v(x) \mid \text{no restrictions}\} \tag{5}$$

4 Weak Form/Virtual Work Equation

For the given problem the virtual work equation reads:

$$\begin{aligned} \int_0^L \delta v''(x) EI v''(x)\, dx &= \int \delta v(x) q(x)\, dx \\ &- \delta v(0)V(0) - \delta v'(0)M(0) \\ &+ \delta v(L)V(L) + \delta v'(L)M(L) \end{aligned} \tag{6}$$

5 Hermite Hat Functions

In the spirit of the linear hat functions which we used before, we will assume an expansion for the displacement and the virtual displacement as:

$$v(x) = \sum_{J=1}^{n} v_J g_J(x) + \theta_J h_J(x)\,, \tag{7}$$

$$\delta v(x) = \sum_{I=1}^{n} \delta v_I g_I(x) + \delta\theta_I h_I(x)\,, \tag{8}$$

$$\tag{9}$$

where as before I, J index a set of nodes that breaks up the domain into a set of $n-1$ elements. The parameters in this setting are v_J and θ_J and they also represent the beam displacement and rotation at the nodes. The functions have the following properties:

1. $g_J(x)$ has unit value at the node it is associated with. It has zero slope there. At the neighboring nodes it has zero value as well as zero slope.

2. $h_J(x)$ is zero at the node it is associated with but it has unit slope there. At the neighboring nodes it has zero value as well as zero slope.

The functions are depicted in Fig. 1. In the graph there is a node at -1, 0, and $+1$. The upper graph shows $g_J(x)$ for the node at 0 and the lower graph $h_J(x)$ for the node at 0. Outside of the domain of the two elements attached to a node the functions are identically equal to zero.

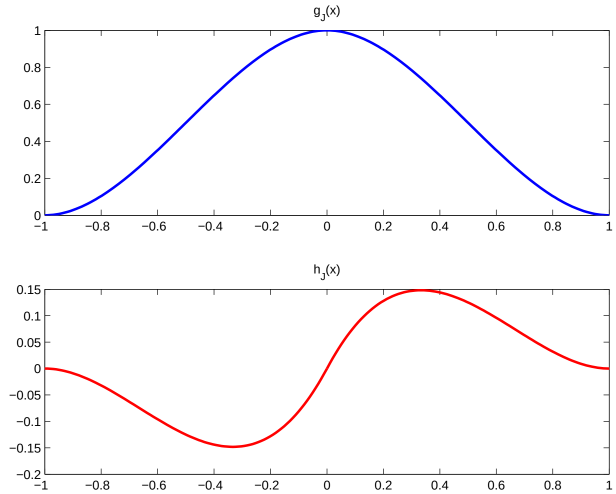

Figure 1: Hermite polynomials for a node located at 0 with neighboring nodes at −1 and +1.

The precise definitions in the general case are given by:

$$g_J(x) = \begin{cases} \frac{3\zeta^2}{\Delta x^2} - \frac{2\zeta^3}{\Delta x^3} & x_{J-1} < x < x_J \\ 1 - \frac{3\xi^2}{\Delta x^2} + \frac{2\xi^3}{\Delta x^3} & x_J < x < x_{J+1} \\ 0 & \text{otherwise}\,, \end{cases} \tag{10}$$

where $\xi = x - x_J$ and $\zeta = x - x_{J-1}$ and

$$h_J(x) = \begin{cases} -\frac{\zeta^2}{\Delta x} + \frac{\zeta^3}{\Delta x^2} & x_{J-1} < x < x_J \\ \xi - \frac{2\xi^2}{\Delta x} + \frac{\xi^3}{\Delta x^2} & x_J < x < x_{J+1} \\ 0 & \text{otherwise}\,. \end{cases} \tag{11}$$

Again, the points $x_J = (J-1)\Delta x$ are the nodes and $\Delta x = L/nel$, where $nel = n - 1$ is the number of elements. Note, as before, the functions on the ends are non-zero over just one element whereas those in the interior are non-zero over two elements.

6 Discrete Equations

The discrete equations are arrived as by plugging our expansions into the virtual work equation, separating out the coefficients of the virtual motion, and noting that the remainder must be zero. The resulting relations are given by

$$\sum_{J=1}^{n} \boldsymbol{K}_{IJ}\boldsymbol{c}_J = \boldsymbol{F}_I \,, \tag{12}$$

where

$$\boldsymbol{K}_{IJ} = \int_0^L EI \begin{bmatrix} g_I'' g_J'' & g_I'' h_J'' \\ h_I'' g_J'' & h_I'' h_J'' \end{bmatrix} dx \tag{13}$$

and

$$\boldsymbol{F}_I = \int_0^L \begin{bmatrix} g_I q \\ h_I q \end{bmatrix} dx - \begin{bmatrix} g_I(0)V(0) \\ h_I'(0)M(0) \end{bmatrix} + \begin{bmatrix} g_I(L)V(L) \\ h_I'(L)M(L) \end{bmatrix} \tag{14}$$

and

$$\boldsymbol{c}_J = \begin{pmatrix} v_J \\ \theta_J \end{pmatrix} . \tag{15}$$

Note that for our particular problem $M(L)$ is known; it is zero. The other end reactions are unknowns and can be found from the 'right-hand side' entries associated with the (three) driven degrees of freedom.

Because the Hermite hat functions have compact support most of the terms in the stiffness matrix are zero. In fact $\boldsymbol{K}_{IJ} = \boldsymbol{0}$ unless $|I - J| \leq 1$. It should also be observed that we already know that $v_1 = 0.0$, $\theta_1 = 0.0$, and $v_n = 0.1$ by the boundary conditions.

To efficiently compute the integrals and implement them in a clean fashion in code, the integrals are computed by computing the contributions from each element and then assembling them into the global stiffness $\boldsymbol{K}_{IJ}$. Thus

$$\boldsymbol{K}_{IJ} = \int_0^L EI \begin{bmatrix} g_I'' g_J'' & g_I'' h_J'' \\ h_I'' g_J'' & h_I'' h_J'' \end{bmatrix} dx = \sum_{A=1}^{nel} \int_{x_A}^{x_{A+1}} EI \begin{bmatrix} g_I'' g_J'' & g_I'' h_J'' \\ h_I'' g_J'' & h_I'' h_J'' \end{bmatrix} dx \,, \tag{16}$$

where *nel* is the total number of elements. If we consider the contribution for a single element with, say nodes A and $A+1$, then the only contributions come from $I, J \in \{A, A+1\}$. This results in four (block) non-zero values for $\boldsymbol{K}_{IJ}$. Thus when computing the contribution from a given element, say element A, we usually form a two-by-two block matrix (four-by-four regular matrix) with block entries:

$$\boldsymbol{k}_{ij}^e = \int_{x_A}^{x_{A+1}} EI \begin{bmatrix} g_i^{e''} g_j^{e''} & g_i^{e''} h_j^{e''} \\ h_i^{e''} g_j^{e''} & h_i^{e''} h_j^{e''} \end{bmatrix} dx \,, \tag{17}$$

where $i, j \in \{1, 2\}$ and $g_1^e = g_A$ and $g_2^e = g_{A+1}$ (similarly for h_i^e). In this way $\boldsymbol{k}_{ij}^e$ is a two-by-two block matrix that contributes to the global stiffness matrix. For example, consider a case where one has 4 elements. Then for element two $\boldsymbol{k}_{11}^{e=2}$ would contribute to as $\boldsymbol{K}_{22} = \boldsymbol{K}_{22} + \boldsymbol{k}_{11}^{e=2}$, $\boldsymbol{K}_{23} = \boldsymbol{K}_{23} + \boldsymbol{k}_{12}^{e=2}$, etc.

We treat the right hand side in a similar fashion; i.e. we form an element right-hand side and assemble it into the global right-hand side. So considering element A, we have

$$\boldsymbol{f}_i^e = \int_{x_A}^{x_{A+1}} \begin{bmatrix} g_i^e q \\ h_i^e q \end{bmatrix} dx - \begin{bmatrix} g_i^e(0)V(0) \\ h_i^{e'}(0)M(0) \end{bmatrix} + \begin{bmatrix} g_i^e(L)V(L) \\ h_i^{e'}(L)M(L) \end{bmatrix}, \tag{18}$$

Here, as before, $i \in \{1, 2\}$ and $g_1^e = g_A$ and $g_2^e = g_{A+1}$ (simlarly for h_i^e). This two-by-one (block) vector is then assembled into the global right-hand side. For example, consider a case where one has 4 elements. Then for element three $\boldsymbol{F}_3 = \boldsymbol{F}_3 + \boldsymbol{f}_1^{e=3}$ and $\boldsymbol{F}_4 = \boldsymbol{F}_4 + \boldsymbol{f}_2^{e=3}$.

6.1 Exercise 2

The block matrix elements $\boldsymbol{k}_{ij}^e$ for a generic element can be computed by hand. Put together, the result is a two-by-two (block) matrix (or four-by-four scalar matrix) and it will be symmetric. The final result is:

$$\boldsymbol{k}^e = \frac{EI}{\Delta x^3} \begin{bmatrix} 12 & 6\Delta x & -12 & 6\Delta x \\ & 4\Delta x^2 & -6\Delta x & 2\Delta x^2 \\ & & 12 & -6\Delta x \\ \text{sym.} & & & 4\Delta x^2 \end{bmatrix}. \tag{19}$$

Verify the first scalar entry of this matrix – i.e. verify that $k_{11}^e = 12EI/\Delta x^3$. For this exercise, it is helpful to note that over a single element there are only 4 non-zero functions. If the first node is at 0 and the second at a, then the two associated with the left most node are:

$$g_{\text{left}}(x) = 1 - \frac{3x^2}{a^2} + \frac{2x^3}{a^3} \tag{20}$$

$$h_{\text{left}}(x) = x - \frac{2x^2}{a} + \frac{x^3}{a^2} \tag{21}$$

and the two associated with the right most node are:

$$g_{\text{right}}(x) = \frac{3x^2}{a^2} - \frac{2x^3}{a^3} \tag{22}$$

$$h_{\text{right}}(x) = -\frac{x^2}{a} + \frac{x^3}{a^2}. \tag{23}$$

6.2 Exercise 3

The block vector entries $\boldsymbol{f}_i^e$ for a generic element can also be computed by hand. The result is a four-by-one scalar vector with the following entries.

$$\boldsymbol{f}^e = \frac{q_o \Delta x}{12} \begin{bmatrix} 6 \\ \Delta x \\ 6 \\ -\Delta x \end{bmatrix} . \tag{24}$$

At the end elements there are contributions from the boundary terms. For element 1, there is the additional contribution

$$\begin{bmatrix} -V(0) \\ -M(0) \\ 0 \\ 0 \end{bmatrix} . \tag{25}$$

For element *nel* (the last element), there is the additional contribution

$$\begin{bmatrix} 0 \\ 0 \\ V(L) \\ M(L) \end{bmatrix} . \tag{26}$$

Verify Eq. (24).

6.3 Exercise 4

Download the file `lab10_student.m` from bspace and complete the program. There is also a plotting program `evalbeam.m` on bspace; once downloaded, type `help evalbeam` to learn how to use.

6.4 Exercise 5

Simplify the loading by setting $q_o = 0$. The problem is then that of a cantilever beam with an end-shear which has the well-known solution. Use this special case to check that your program is correct. You should get an exact answer for any number of elements (even just one element). Look at all aspects of the solution. Forces, moments, displacements, and rotations to make sure that they are correct.

6.5 Exercise 6

Verify that your program converges by checking that the displacement solution converges as the number of nodes increases when you turn the distributed loading back on. Note that with these approximation functions one can increase the number of parameters without the difficulties that arise with our simple polynomials from the earlier labs.

6.6 Exercise 7

Where does the maximum bending moment occur for the problem? and what is its value?

6.7 Exercise 8

Add a mid-span pin support and solve the problem with your code. Where does the maximum bending moment occur now? and what is its value?

6.8 Extra Credit

Create a modified program that solves the buckling problem using these approximation functions and re-compute the answers to some of the prior lab and homework questions. [Note: Extra credit can only be obtained, if you have already completed the first 8 Exercises.]