Before you decide to drop the course and curse me for convincing or making you take this class, let me tell you that up till now we have not really discussed finite element methods.

The basic mathematics we have previously discussed provides us with a basis for understanding the inner workings of FEM techniques. The two basic approaches are:

1. The Ritz method - utilizes the energy or functional associated with the differential equation as the base for the finite element formulation
2. The MWR method - approximates the differential equation directly as the base for the finite element formulation

The steps involved in generating a FEM model using variational techniques follows the same procedure as we used in our discussion of the spring-mass system:

1. Discretization
2. Interpolation
3. Elemental Description or Formulation
4. Assembly
5. Constraints
6. Solution
7. Computation of Derived Variables

Discretization - The domain is broken up into a series of subintervals:

The points that separate the subintervals are called nodes and the subdomains between the nodes are called elements.

The entire discretization is often called a mesh. Later we will discuss how the spacing of the mesh may effect the solution.

Matching the values of the function at the endnodes of each element require $u_e$ to be:

$$u_x = c_1 + c_2 x$$

Matching the values of the function at the endnodes of each element require $u_x$ to be:

$$u_x = c_1 + c_2 x$$

The variation of the unknown function over an element may be written in global coordinates as:

$$u_x = c_i = C_i X$$

If for example, we are interested in information based on the derivative at the nodes, then a linear interpolation will not be appropriate for the problem.

An element of this type assumes a linear variation of the function over the element.

Therefore the function is represented as a series of linear lines that are continuous at the nodes.

One obvious problem with a linear interpolation is that the derivative of the approximation is discontinuous at each node.

Interpolation - At this point we have to decide what type of interpolation or approximation we want over each element.

In past discussions of the axial deformation problem we routinely used a linear element.

The points that separate the subintervals are called nodes and the subdomains between the nodes are called elements.
Solving for \( c_1 \) and \( c_2 \) results in the following equation:

\[
\begin{align*}
    u_e &= u \left( \frac{x_{i,1} - x}{x_{i,1} - x_j} \right) + u_{i,1} \left( \frac{x - x_{1,1}}{x_{1,1} - x_j} \right) \\
    u_e &= N_i u_i + N_{i,1} u_{i,1}
\end{align*}
\]

where:

\[
N_i = \frac{x_{i,1} - x}{x_{i,1} - x_j} \quad \text{and} \quad N_{i,1} = \frac{x - x_{1,1}}{x_{1,1} - x_j}
\]

The derivative of \( u \) may be computed as:

\[
\begin{align*}
    u_e' &= u \left( \frac{-1}{x_{i,1} - x_j} \right) + u_{i,1} \left( \frac{1}{x_{1,1} - x_j} \right) \\
    u_e' &= N_i' u_i + N_{i,1}' u_{i,1}
\end{align*}
\]

where:

\[
N_i' = -\frac{1}{x_{i,1} - x_j} \quad \text{and} \quad N_{i,1}' = \frac{1}{x_{1,1} - x_j}
\]

In matrix form we may write the variation of \( u \) and \( u' \) over an element as:

\[
\begin{align*}
    u_e &= N^T u_s \quad \text{and} \quad u_e' = N'^T u_s
\end{align*}
\]

where the vectors \( N \), \( N' \), and \( u_s \) are:

\[
N = \begin{bmatrix} N_i \\ N_{i,1} \end{bmatrix} \quad N' = \begin{bmatrix} N_i' \\ N_{i,1}' \end{bmatrix} \quad u_s = \begin{bmatrix} u_i \\ u_{i,1} \end{bmatrix}
\]

The vector \( N \) is often called the elemental interpolation vector and \( u_s \) is called the elemental displacement vector.

It is often more convenient to express the interpolation functions in terms of an elemental coordinate system \( \zeta \).

For example, the linear interpolation functions may be rewritten in \( \zeta \) as:
These elemental interpolation functions are often called **shape functions** because they determine the shape of the unknown function over each element.

The derivative shape functions are:


Recall the energy functional is:

\[
Z(u) = \int_a^b \left( p(u')^2 - qu^2 - uf \right) dx + \frac{\alpha u(a)^2}{2} + \frac{\beta u(b)^2}{2} + Au(a) + Bu(b)
\]

Using the concept of the **element** as we have discussed previously, we may approximate the functional by replacing the continuous function \( u \) with a series of connected elements.

Each of these integrals is evaluated over each element.

To transform the above integral into element coordinates \( \xi \) the differential operator \( dx \) must be replaced by the appropriate transformation between \( x \) and \( \xi \).

The global coordinate \( x \) may be written as:

\[
x_e = N^T x_n = \sum_{i=1}^{N_e} N_i x_i
\]

Differentiating \( x \) with respect to \( \xi \) gives:

\[
dx \frac{dx}{d\xi} d\xi = \frac{dx}{d\xi} \left[ (1-\xi) x_i + \xi x_{i+1} \right] d\xi = (x_{i+1} - x_i) d\xi
\]

\[
dx = l_e d\xi
\]

\[
x_e = x_i + \xi (x_{i+1} - x_i) \Rightarrow \xi = \frac{x_e - x_i}{l_e}
\]

\[
du \frac{du}{dx} \frac{dx}{d\xi} d\xi = du \frac{d}{d\xi} \frac{x - x_i}{l_e} \Rightarrow \frac{du}{dx} \frac{1}{l_e}
\]
FINITE ELEMENT FORMULATIONS

Element Formulation - Substituting the coordinate transformation for \( x \) in the integrals \( Z_{pe}, Z_{qe}, \) and \( Z_{fe} \) result in:

\[
Z_{pe} = \frac{1}{I_p} \int_0^1 u' p(x) u l_0 d \xi
\]

\[
Z_{qe} = \frac{1}{I_q} \int_0^1 q(x) l_0 d \xi
\]

\[
Z_{fe} = \frac{1}{I_f} \int_0^1 f(x) l_0 d \xi
\]

Now we replace the function \( u \) and its derivative \( u' \) with the linear elemental approximation using the shape functions in the elemental coordinate \( \xi \). For example, consider the integral \( Z_{pe} \):

\[
Z_{pe} = \frac{1}{I_p} \int_0^1 \hat{u}^T p(x) \hat{N}^T u_\alpha l_0 d \xi
\]

where \( p_\alpha \) is defined as:

\[
p_\alpha = \int_0^1 \hat{N}^T p(x) \hat{N} d \xi
\]

FINITE ELEMENT FORMULATIONS

Element Formulation - Let’s examine in detail the integrals \( p_\alpha, q_\alpha, \) and \( f_\alpha \). Consider the integral \( p_\alpha \):

\[
p_\alpha = \int_0^1 \hat{N}^T p(x) \hat{N} d \xi = \int_0^1 \left[ \begin{array}{c} N_i \\ N_{i+1} \end{array} \right] p(x) \left[ \begin{array}{c} N_i \\ N_{i+1} \end{array} \right]^T d \xi
\]

\[
= \frac{1}{I_p} \left[ \begin{array}{c} 1 \\ -1 \end{array} \right] p(x) \left[ \begin{array}{c} -1 \\ -1 \end{array} \right] d \xi
\]

\[
x = x_i + \xi l_i
\]

FINITE ELEMENT FORMULATIONS

Element Formulation - Now let us consider the \( f_\alpha \) integrals:

\[
f_\alpha = \int_0^1 \hat{N}^T f(x) l_0 d \xi = \int_0^1 \left[ \begin{array}{c} N_i \\ N_{i+1} \end{array} \right] f(x) l_0 d \xi
\]

\[
f_\alpha = \int_0^1 \left[ \begin{array}{c} 1 \\ -1 \end{array} \right] f(x_i + \xi l_i) l_0 d \xi
\]
Assembly - The functional \( Z(u) \), through the discretization and interpolation procedures, has been converted into an approximate function \( Z(u_1, u_2, u_3, \ldots, u_{N+1}) \), which may be written as:

\[
Z(u_1, u_2, u_3, \ldots, u_{N+1}) = \sum \frac{u_i^T k_i u_i}{2} - \sum u_i^T f_i
\]

where \( k_i = p_i + q_i \). Consider the \( k_i \) term from the first element. On an element level, the term \( u_i^T k_i u_i \) has the form:

\[
u_i^T k_i u_i = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
\]

Assembly - In a similar manner, the contribution of the forcing function \( f \) from the first element may be written as:

\[
u_i^T f_i = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} f_i \\ f_i \end{bmatrix}
\]

The global contribution of the first element is:

\[
u_i^T f_i = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} f_i \\ f_i \end{bmatrix} = u_i^T f_i
\]

Assembly - Recall the energy functional \( Z(u) \) has the form:

\[
Z(u_1, u_2, u_3, \ldots, u_{N+1})
\]

and has a stationary value that is obtained by requiring each partial derivative to vanish:

\[
\frac{\partial Z}{\partial u_i} = 0 \quad i = 1, 2, \ldots, N + 1
\]

\[
\frac{\partial Z}{\partial u_0} = 0 = \frac{\partial}{\partial u_0} \left( \frac{u_0^T K_0 u_0}{2} - u_0^T F_0 \right) \rightarrow \begin{bmatrix} K_0 u_0 = F_0 \end{bmatrix}
\]

We have converted the original functional into a function of the nodal unknowns and then required the functional to be stationary with respect to each node using the Ritz method.

Assembly - Substituting this element “stiffness” matrix into the global system would result in something like this:

\[
u_i^T k_i u_i - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = u_i^T k_i u_i
\]

The above equation is intended to show how each element contributes to the global system.
### Constraints
- If $C$ is zero then the first column in the "stiffness" matrix may also be replace by zeros.
- However, for non-zero values of $C$ we cannot replace the stiffness components in the first column.
- The unconstrained system equations were originally represented by a symmetric stiffness matrix.
- With boundary conditions accounted for, the stiffness matrix has become unsymmetrical.

### Solution
- The equations are ready to be solved.
- Since we have spent some energy symmetrizing the global equations, we should use an equation solver that uses a symmetric storage algorithm.

### Computation of Derived Variables
- For the Sturm-Liouville problem we have been discussing, the quantity $u'$ is not solved for directly.

### Discretization
- The domain is broken up into a series of subintervals:
- The points that separate the subintervals are called **nodes** and the subdomains between the nodes are called **elements**.
- The entire discretization is often called a **mesh**. Later we will discuss how the spacing of the mesh may effect the solution.
Interpolation - At this point we have to decide what type of interpolation or approximation we want over each element.

In past discussions of variational model we used a linear element. An element of this type assumes a linear variation of the function over the element.

For a Galerkin FEM model we will reexamine the concept of interpolation over an element.

In developing a Galerkin FEM model, the solution is represented in the form of a set of admissible functions:

\[ u = \sum_{i=1}^{N+1} u_i n_i(x) \]

where each of the function \( n_i(x) \) is composed of parts of the adjacent interpolation functions \( N_i(x) \).

The functions \( n_i(x) \) may be called the basis functions for the piecewise linear approximation over the interval \([a, b]\).

Substituting our approximation of the function \( u \) into the above expression from the weighted residual or the error statement:

\[ E_{w}(U) = \int_a^b \left( \nu' p + v q \right) n_i \left( u n_i' - v q n_i \right) dx + \alpha \nu \left( a u(a) + \beta \nu(b) u(b) \right) \]

\[ = \int_a^b \left( \nu f + a u + B v \right) n_i dx = 0 \]

The Galerkin MWR model requires that the sum of the residual \( E_{w}(U) \) be zero.

Element Formulation - If we take the weighting functions \( \nu(x) \) as the basis function then the MWR statement becomes:

\[ \sum_{i=1}^{N+1} \int_a^b \left( n_i' p - n_i q \right) u_i dx + \alpha \nu \delta_{i1} + \beta \nu \delta_{iN+1} = 0 \]

where \( \delta_{ij} \) is the Kronecker delta function given as:

\[ \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \]

which implies that \( \delta_{ij} \) term occurs only during the \( N+1 \) equation and \( \delta_{ij} \) during the first equation.
Element Formulation - The approximation equations may be written in matrix form as:
\[ \sum A_k u_k = b_k \quad k = 1, 2, \ldots, N + 1 \rightarrow \mathbf{A} \mathbf{u} = \mathbf{b} \]

where the terms \( A_k \) and \( b_k \) are defined as:
\[
A_k = \sum_{n=1}^{N} \left( n'_p n'_q - n_q n_p \right) dx + \alpha u_{N+1} + \beta u_{N+2} \delta_{N+1} \delta_{N+2} \\
b_k = \sum_{n=1}^{N} \left( n_i f \right) dx + A \delta_{N+1} + B \delta_{N+2}
\]

Element Formulation - Therefore, the \( k \)th equation corresponding to the node at \( x_k \) reduces to:
\[
\sum_{n=1}^{N} \left( \int_{x_{n-1}}^{x_n} \left( n'_p n'_q - n_q n_p \right) dx + u_{n-1} \int_{x_{n-1}}^{x_n} \left( n'_p n'_q - n_q n_p \right) dx \right) u_{n-1} = \int_{x_{n-1}}^{x_n} \left( n_i f \right) dx
\]

Let's look a little closer at the equation associated with the \( n \)th node:
\[
u_{n-1} \int_{x_{n-1}}^{x_n} \left( n'_p n'_q - n_q n_p \right) dx + u_{n-1} \int_{x_{n-1}}^{x_n} \left( n'_p n'_q - n_q n_p \right) dx + u_{n+1} \int_{x_{n-1}}^{x_n} \left( n'_p n'_q - n_q n_p \right) dx = \int_{x_{n-1}}^{x_n} \left( n_i f \right) dx
\]

Notice that \( n_{k-1} \) is zero over the interval \([x_{k-1}, x_k]\) and \( n_{k+1} \) is zero over the interval \([x_k, x_{k+1}]\).

Element Formulation - Since \( n_{k-1} \) and \( n'_{k+1} \) are zero over the interval \([x_{k-1}, x_k]\) and \( n_{k+1} \) and \( n'_k \) are zero over the interval \([x_k, x_{k+1}]\) then the above equation becomes:
\[
u_{k-1} \int_{x_{k-1}}^{x_k} \left( n'_p n'_q - n_q n_p \right) dx + u_{k-1} \int_{x_{k-1}}^{x_k} \left( n'_p n'_q - n_q n_p \right) dx + u_{k+1} \int_{x_{k-1}}^{x_k} \left( n'_p n'_q - n_q n_p \right) dx = \int_{x_{k-1}}^{x_k} \left( n_i f \right) dx
\]

Do these functions look familiar?
**FINITE ELEMENT FORMULATIONS**

**Element Formulation** - In order to compare this formulation with the Ritz approach we discussed previously, let’s write out a set of equations for $N+1 = 3$ as:

\[
\begin{align*}
F_1 &= n'_1 p_{n_1} + n_1 q_{n_1} \\
F_2 &= n'_2 p_{n_2} + n_2 q_{n_2} \\
F_3 &= n'_3 p_{n_3} + n_3 q_{n_3}
\end{align*}
\]

**FINITE ELEMENT FORMULATIONS**

**PROBLEM #11** - Complete the verification that the equations of the Galerkin approach coincide with the equations derived from the Ritz approach.

**Hint:** First write the Galerkin method in elemental coordinates $\xi$ and then determine if the elemental stiffness components are comparable and if the terms associated with the function $f(x)$ are identical.

---

**Numerical Integration**

Before we attempt to solve some problem using the Ritz or Galerkin FEM formulations, we need to discuss how to evaluate integrals of the form:

\[
I = \int_0^1 f(\xi) \, d\xi
\]

The basic idea behind any approximate integrations or quadrature method is to replace the actual function with a polynomial that accurately estimates the behavior of the integrand.

The form of the approximate functions is such that their exact integration may be easily computed.

**Trapezoidal Rule** - In this method the function $f(\xi)$ is replaced by a series of connected linear functions $F(\xi)$. The integral may be estimated as:

\[
I = \sum_{i=1}^{N} \left[ F(\xi_i) + F(\xi_{i+1}) \right] \frac{h_i}{2}
\]

where $F$ is a linear function over the interval $[\xi_i, \xi_{i+1}]$. Computing the area of the trapezoidal region under the line $F$ over each subinterval results in:

\[
I = \sum_{i=1}^{N} \left[ F(\xi_i) + F(\xi_{i+1}) \right] \frac{h_i}{2}
\]

**Simpson's Rule** - In this method the function $f(\xi)$ is replaced by a series of connected quadratic functions $F(\xi)$.

The quadratic function is defined over two subintervals of length $2h_i$.

The approximated quadratic curve may be integrated directly over both subintervals.

The resulting value of the integral is:

\[
I = \sum_{i=1}^{N} \left[ F(\xi_i) + 4F(\xi_{i+1}/3) + F(\xi_{i+1}) \right] \frac{h_i}{3}
\]

\[
h_i = \frac{\xi_{i+1} - \xi_i}{2}
\]
**Numerical Integration**

**Simpson’s Rule** - If all the subintervals are assumed to be the same size, then the integral may be written as:

\[
I = \frac{b-a}{3} \left( F(a) + 4 \sum_{i=1,3,5,...}^{N} F(x_i) + 2 \sum_{i=2,4,6,...}^{N} F(x_i) + F(b) \right)
\]

This formula is often called Simpson’s 1/3 Rule.

**Gauss-Legendre Quadrature** - In this method, a series of specialized points and weights are developed to evaluate the integral.

We will briefly discuss how these points and weights are determined.

The general form of the quadrature is:

\[
I = \int_{-1}^{1} F(u) \, du = \sum_{i=1}^{N} w_i F(u_i)
\]

**Gauss-Legendre Quadrature** - A general integral:

\[
I = \int_{a}^{b} F(\xi) \, d\xi
\]

may be converted into the Gaussian integration space \([-1, 1]\) by the following transformation:

\[
\xi = \frac{b-a}{2} \left( \frac{b-a}{2} \right) u \quad d\xi = \frac{b-a}{2} \, du
\]

Therefore the integral has the form:

\[
I = \int_{-1}^{1} F \left( \frac{b-a}{2} + \frac{(b-a)u}{2} \right) \left( \frac{b-a}{2} \right) \, du
\]

**Gauss-Legendre Quadrature** - Let’s examine how the points and weights for Gaussian quadrature are developed. Assume a one-point method, therefore \(N = 1\):

\[
I = \int_{-1}^{1} F(u) \, du = w_i F(u_i)
\]

The quadrature point \(u_i\) and weight \(w_i\) are determined such that any linear function of \(u\) may be integrated exactly.

**Gauss-Legendre Quadrature** - We can accomplish this task by solving the following two equations for \(u_i\) and \(w_i\):

\[
\begin{align*}
I &= \int_{-1}^{1} du = 2w_i F(u_i) = w_i \\
I &= \int_{-1}^{1} u \, du = 0 = w_i \mu_i \quad \therefore \mu_i = 0
\end{align*}
\]

Thus the one-point Gaussian quadrature takes the form:

\[
I = \int_{-1}^{1} F(u) \, du = 2F(0)
\]
Gauss-Legendre Quadrature - To determine the points and weights for a two-point Gaussian quadrature, \( N = 2 \), the following equations should be solved:

\[
F = 1 \int_1^1 1 \, du = w_1 F(u_1) + w_2 F(u_2) = w_1 + w_2 \\
F = u \int_1^1 u \, du = 0 = w_1 F(u_1) + w_2 F(u_2) = w_1 u_1 + w_2 u_2 \\
F = u^2 \int_1^1 u^2 \, du = \frac{2}{3} = w_1 F(u_1) + w_2 F(u_2) = w_1 u_1^2 + w_2 u_2^2 \\
F = u^3 \int_1^1 u^3 \, du = 0 = w_1 F(u_1) + w_2 F(u_2) = w_1 u_1^3 + w_2 u_2^3
\]

Thus the two-point Gaussian quadrature takes the form:

\[
F(u) du = F\left(-\frac{1}{\sqrt{3}}\right) + F\left(\frac{1}{\sqrt{3}}\right)
\]

Therefore, a two-point Gaussian quadrature formula will exactly integrate up to a third-order polynomial.

Gauss-Legendre Quadrature - Solving this set of equations gives:

\[
\begin{align*}
111 & = -1 \\
0 & = 1 \\
\frac{2}{3} & = 1 \\
0 & = -1
\end{align*}
\]

Thus the two points are:

\[
\begin{align*}
&u_1 = -\frac{1}{\sqrt{3}} \approx -0.577 \\
&u_2 = \frac{1}{\sqrt{3}} \approx 0.577
\end{align*}
\]

The weights are:

\[
\begin{align*}
w_1 & = w_2 = \frac{1}{\sqrt{3}} \approx 0.577
\end{align*}
\]

Gauss-Legendre quadrature has the property that an \( N \)th order approximation integrates exactly a polynomial of degree \( 2N-1 \) or less.

**PROBLEM #12** - To compare and contrast the different integration methods we have discussed, evaluate the following integral:

\[
I = \int_0^1 \frac{\sin(x)}{1 + x^2} \, dx
\]

by:
1. trapezoidal rule with one, two, and four intervals;
2. Simpson’s rule using two and four intervals;
3. Gaussian quadrature for \( N = 1 \) to 6.

**Example** - Consider the problem of the axial deformation of a prismatic bar we worked previously.

The boundary value problem for this case is:

\[
(\frac{AE}{l})' x + Q(x) = 0 \quad 0 \leq x \leq L
\]

the boundary conditions are:

\[
\begin{align*}
&u(0) = 0 \\
&AEu'(L) = 0
\end{align*}
\]

**Example** - The Sturm-Liouville form of this equation requires that \( p = AE \), \( q = 0 \), and \( A = B = \alpha = \beta = 0 \). The corresponding functional is:

\[
Z(u) = \int_0^L \frac{(AEu)'^2}{2} - Q(x)u \, dx
\]

**Discretization** - The domain will be divided into four elements.
**FINITE ELEMENT FORMULATIONS**

**Interpolation** - We will use linear elements. In developing a Ritz FEM model, the solution was represented in the form of a set of admissible functions:

\[ u = N_1 u_1 + N_2 u_2 + \cdots + N_{n+1} u_{n+1} \]

where \( x = N_1 x_1 + N_2 x_2 + \cdots + N_{n+1} x_{n+1} \)

**Element Formulation** - The approximation of the energy functional may be written in the following form:

\[ Z(u) = \int_0^1 \left( \frac{1}{2} \left( \frac{\partial u}{\partial \xi} \right)^2 - Z_0 \right) dx \]

where:

\[ Z_0 = \int_0^1 \frac{1}{2} u^T A E u \, d\xi \]

**Element Formulation** - Now we replace \( u' \) with the linear elemental approximation using the shape functions in the elemental coordinate \( \xi \).

For example, consider the integral \( Z_{pe} \):

\[ Z_{pe} = \int_0^1 u^T A E u \, d\xi \]

**Assembly** - Consider the \( k_0 \) term for each element in the system.

For element #1: \( x = 0 \)

\[ k_1 u_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \]

For element #2: \( x = 2.5 \) ft.

\[ k_2 u_2 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} \]

For element #3: \( x = 5 \) ft.

\[ k_3 u_3 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} \]

For element #4: \( x = 7.5 \) ft.

\[ k_4 u_4 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_4 \\ u_5 \end{bmatrix} \]

**Assembly** - The right-hand side terms involving the loading function for each element are:

\[ f_e = \int_0^1 \left( 1 - x \right) \frac{1 - \xi}{L} \, d\xi = \begin{bmatrix} \frac{Q_0}{6L}(-3x_i - 1) \\ \frac{Q_0}{6L}(-3x_i - 2L + 1) \end{bmatrix} \]

for \( l_e = L/4 \) the elemental integrals become:

\[ f_e = \begin{bmatrix} \frac{Q_0}{96} \left( 1L - 12x_i \right) \\ \frac{Q_0}{96} \left( 10L - 12x_i \right) \end{bmatrix} \]
**FINITE ELEMENT FORMULATIONS**

**Assembly** - Consider the \( f \) term for each element in the system.

For element #1: \( x = 0 \)
\[
f_1 = \begin{bmatrix} 11 QL \ \ 10 QL \end{bmatrix} \frac{96}{1}.
\]

For element #2: \( x = 2.5 \text{ ft.} \)
\[
f_2 = \begin{bmatrix} 8 QL \ \ 7 QL \end{bmatrix} \frac{96}{1}.
\]

For element #3: \( x = 5 \text{ ft.} \)
\[
f_3 = \begin{bmatrix} 5 QL \ \ 4 QL \end{bmatrix} \frac{96}{1}.
\]

For element #4: \( x = 7.5 \text{ ft.} \)
\[
f_4 = \begin{bmatrix} 2 QL \ \ QL \end{bmatrix} \frac{96}{1}.
\]

\[ \text{Assembly} - \text{Compiling these terms into the global system equations gives:} \]
\[
K_u u = F_0
\]

\[
= \frac{4AE}{L}
\]

**FINITE ELEMENT FORMULATIONS**

**Constraints** - Essential or forced boundary conditions may be inserted into the system equations by simply rewriting the appropriate equation corresponding to the variable where the condition is prescribed.

The global system “stiffness” matrix constrained by the boundary condition \( u(0) = 0 \) is:
\[
\begin{bmatrix}
1 & -1 & 0 & 0 & 0 \\
-1 & 2 & -1 & 0 & 0 \\
0 & -1 & 2 & -1 & 0 \\
0 & 0 & -1 & 1 & 1
\end{bmatrix}
\]

The exact solution may be determined from the following expression:
\[
AEu(x) = \int_0^x Q(x) dx^* dx = \int_0^x \left( 1 - \frac{x}{L} \right) dx^* dx
\]

\[
= QL \left[ \frac{1}{6} \left( \frac{x}{L} \right)^3 - \frac{1}{2} \left( \frac{x}{L} \right)^2 + \frac{1}{2} \left( \frac{x}{L} \right) \right]
\]

**Solution** - The exact solution may be determined from the following expression:
\[
\begin{align*}
&u_1 = 0 \quad u_2 = 0.4784 \text{ in.} \quad u_3 = 0.7241 \text{ in.} \\
u_4 = 0.8147 \text{ in.} \quad u_5 = 0.8276 \text{ in.}
\end{align*}
\]

**FINITE ELEMENT FORMULATIONS**

**Solution** - The exact solution may be determined from the following expression:
\[
\begin{align*}
&u_1 = 0 \quad u_2 = 0.4784 \text{ in.} \quad u_3 = 0.7241 \text{ in.} \\
u_4 = 0.8147 \text{ in.} \quad u_5 = 0.8276 \text{ in.}
\end{align*}
\]
**FINITE ELEMENT FORMULATIONS**

**FEM Solution** - Comparison of 4 element FEM formulation with exact solution

![Graph showing comparison of FEM solution with exact solution](image)

**FEM Solution** - For the Sturm-Liouville problem we have been discussing, the quantity \( u' \) is not solved for directly.

Since we used a linear interpolation in the variational formulation we can calculate an approximate value of \( u' \) using the linear elemental interpolation functions:

\[
u'_e = \frac{u_{i+1} - u_i}{l_e}
\]

**FEM Solution** - Substituting the numerical values for \( Q_0, L, A, \) and \( E \) into the displacement expressions gives:

\[
u'_{element 1} = -37 \frac{QL}{96AE} \quad \nu'_{element 2} = -19 \frac{QL}{96AE} \quad \nu'_{element 3} = -6 \frac{QL}{96AE} \quad \nu'_{element 4} = -\frac{QL}{96AE}
\]

**Computation of Derived Variables** - The exact solution for the first derivative is:

\[
AEu'(x) = -\int Q(x) dx' dx = -\int Q(x) \left(1 - \frac{x}{L}\right) dx' dx
\]

\[
= QL \left[\frac{1}{2} \left(\frac{1}{2}x^2\right) - \frac{1}{2} \left(\frac{x}{L}\right) + \frac{1}{2} \left(\frac{x}{L}\right)^2\right]
\]

**FEM Solution** - Substituting the numerical values for \( Q_0, L, A, \) and \( E \) into the displacement expressions gives:

\[
u'_{element 1} = -37 \frac{QL}{96AE} \quad \nu'_{element 2} = -19 \frac{QL}{96AE} \quad \nu'_{element 3} = -6 \frac{QL}{96AE} \quad \nu'_{element 4} = -\frac{QL}{96AE}
\]

**Computation of Derived Variables** - The exact solution for the first derivative is:

\[
u'_1 = 48 \frac{Q_0L}{96AE} \quad \nu'_2 = 27 \frac{Q_0L}{96AE} \quad \nu'_3 = 12 \frac{Q_0L}{96AE}
\]

\[
u'_4 = 3 \frac{Q_0L}{96AE} \quad \nu'_5 = 0
\]

**FEM Solution** - Substituting the numerical values for \( Q_0, L, A, \) and \( E \) into the displacement expressions gives:

\[
u'_{element 1} = -37 \frac{QL}{96AE} \quad \nu'_{element 2} = -19 \frac{QL}{96AE} \quad \nu'_{element 3} = -6 \frac{QL}{96AE} \quad \nu'_{element 4} = -\frac{QL}{96AE}
\]

**Example** - Work the preceding problem again using eight equally-spaced elements.

The boundary value problem for this case is:

\[
(AEu')' + Q(x) = 0 \quad 0 \leq x \leq L
\]

the boundary conditions are:

\[
\begin{align*}
u(0) &= 0 \\
AEu'(L) &= 0
\end{align*}
\]
**Example** - Work the preceding problem again using eight equally-spaced elements.

**Discretization** - The domain will be divided into eight elements.

\[ u(x) \]

\[ \frac{x}{L} \]

\[ \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix} = \frac{Q L^2}{3072AE} \]

\[ \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix} = \frac{Q L^2}{3072AE} \]

\[ \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix} = \frac{Q L^2}{3072AE} \]

**Example** - Applying the values for the geometry, material properties, and loading given in this problem and the boundary condition results in:

\[ \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix} = \frac{Q L^2}{3072AE} \]

\[ \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \end{bmatrix} = \frac{Q L^2}{3072AE} \]

**Example** - The solution of these equations is:

\[ u_1 = 0 \quad u_3 = 296 \quad u_5 = 448 \]

\[ u_7 = 504 \quad u_9 = 512 \]

\[ QL \quad A \quad E \]

\[ u_1 = 0 \quad u_3 = 0.4784 \text{ in.} \quad u_5 = 0.7241 \text{ in.} \]

\[ u_7 = 0.8147 \text{ in.} \quad u_9 = 0.8276 \text{ in.} \]

**Computation of Derived Variables** - The exact solution for the first derivative is:

\[ u'_1 = 169 \quad u'_3 = 91 \quad u'_5 = 37 \quad u'_7 = 7 \]

\[ u'_9 = \frac{Q L}{384AE} \]

**Example** - Substituting the numerical values for \( Q_0, L, A, \) and \( E \) results in:

\[ u_1 = 0 \quad u_3 = 0.4784 \text{ in.} \quad u_5 = 0.7241 \text{ in.} \]

\[ u_7 = 0.8147 \text{ in.} \quad u_9 = 0.8276 \text{ in.} \]

**FEM Solution** - Comparison of 8 element FEM formulation with exact solution

\[ u(x) = \frac{Q L}{384AE} \]

**FEM Solution** - Comparison of 8 element FEM formulation with exact solution

\[ u'(x) = \frac{Q L}{384AE} \]
PROBLEM #13 - Correct the "bleeped" version of ODE2 - linear second-order differential equation solver.

Use the ODE2 program to solve following axial deformation of the prismatic bar:

\[ u(x) = Q(x) \left( 1 - \frac{x}{L} \right) \]

\[ Q(x) = Q_0 \left( 1 - \frac{x}{L} \right) \]

- \( E = 29,000 \text{kpsi} \)
- \( A = 1 \text{in.}^2 \)
- \( Q_0 = 10 \text{kips/in.} \)

10 ft.