Finite Element Approaches

December 4, 2017

Finite Element Solutions of Boundary-value Problems in ODEs
Larry Caretto
Mechanical Engineering 501AB
Seminar in Engineering Analysis
December 4, 2017

Outline
• Review last week on eigenvalue problems with ordinary differential equations
• Finite element methods for boundary value problems
  – Elements and shape (basis) functions
  – Getting the numerical equations by method of weighted residuals
  – Example problem and results
  – Comparison with finite difference approach

Review Eigenvalue Problem
• Numerical eigenvalue problems occur in ODEs when the number of boundary conditions is greater than the order of the differential equation
  – Example of this is solution for burning velocity of a laminar flame
• Basic approach is to use finite-differences and transform problem into a numerical matrix eigenvalue problem

Review Eigenvalue Problem II
• Look at simple problem with known solution as an example
  – \( \frac{d^2y}{dx^2} + \lambda^2 y = 0 \) with \( y(0) = 0, y(1) = 0 \) and \( \int y dx = 1 \)
  – Have three boundary conditions and only a second order equation
  – Known solution is \( y = A \sin \lambda x \) with \( \lambda = n\pi \)
• Use second order finite differences
  – \( (y_{i+1} + y_{i-1} - 2y_i)h^2 + \lambda^2 y_i = 0 \)

Review Eigenvalue Problem III
• Have matrix eigenvalue problem with \( \alpha = -\lambda^2 h^2 \) as the eigenvalue

Review Eigenvalue Problem IV
• Solve by numerical techniques for finding matrix eigenvalues
• The accuracy of the eigenvalues depends on the grid
• Often need only one (lowest or highest)
• Can only find as many eigenvalues as there are grid nodes (not counting boundary nodes)
Review Eigenvalue Problem V

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Numerical</th>
<th>Exact</th>
<th>Percent error</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.090</td>
<td>3.142</td>
<td></td>
<td>1.66%</td>
</tr>
<tr>
<td>5.878</td>
<td>6.283</td>
<td></td>
<td>6.45%</td>
</tr>
<tr>
<td>8.090</td>
<td>9.425</td>
<td></td>
<td>14.16%</td>
</tr>
<tr>
<td>9.511</td>
<td>12.566</td>
<td></td>
<td>24.31%</td>
</tr>
</tbody>
</table>

Finite Element Methods

• Designed for 2D and 3D geometries
• Can use for 1D case as example
• Basic idea is to divide region into small elements (line, area, volume)
• Use interpolating polynomial for each element
  – Represent both geometry (independent variables) and dependent variable
  – Interpolating polynomials called basis functions

Finite Element Methods II

• Analysis for individual elements is assembled into a set of nodal equations for the entire region
  – Result is set of algebraic equations for the dependent variable at nodes that are points on elements
  – Because we do not use coordinate lines for grid, it is easier to model complex engineering geometries
  – Finite-volume methods for fluid calculations

Two-dimensional Element

\[
\begin{align*}
(x_4, y_4) & \quad \xi = -1, & \xi = 1, \\
(x_3, y_3) & \quad \eta = 1, & \eta = 1 \\
(x_2, y_2) & \quad \xi = -1, & \xi = 1, \\
(x_1, y_1) & \quad \eta = -1 & \eta = -1
\end{align*}
\]

• Use dimensionless \( \xi, \eta \) coordinate system for basis functions
• Each element has several shape or basis functions, \( \phi_i \)

Shape (Basis) Functions

• Model geometry and dependent variable over the element
• Use of same basis functions for both is called isoparametric element
• Shape functions associated with element nodes such that \( \phi_i(x_i) = \delta_{ij} \)

\[
x = \sum_{i=1}^{4} x_i \phi_i, \quad y = \sum_{i=1}^{4} y_i \phi_i, \quad T = \sum_{i=1}^{4} T_i \phi_i
\]

Shape (Basis) Functions II

• Simplest shape functions are linear for 1D or bilinear for 2D
• For a linear element between nodes \( i \) (at \( \xi = -1 \)) and \( i + 1 \) (at \( \xi = 1 \)) we have \( \phi_i = (1 - \xi)/2 \) and \( \phi_{i+1} = (1 + \xi)/2 \)
• \( x = x_i \phi_i + x_{i+1} \phi_{i+1} \) is correct at 1D nodes
• Bilinear functions for 2D element have the form \((1 \pm \xi)/(1 \pm \eta)/2\)
Bilinear Shape Functions

\[
\begin{align*}
\xi &= -1, & \eta &= 1, \\
\xi &= 1, & \eta &= 1 \\
\xi &= -1, & \eta &= -1 \\
\xi &= 1, & \eta &= -1 \\
\phi_1 &= \frac{(1-\xi)(1-\eta)}{4}, & \phi_2 &= \frac{(1+\xi)(1-\eta)}{4}, \\
\phi_3 &= \frac{(1+\xi)(1+\eta)}{4}, & \phi_4 &= \frac{(1-\xi)(1+\eta)}{4}
\end{align*}
\]

Modeling Differential Equation

- Look at same example used for finite differences: \(\frac{d^2T}{dx^2} + a^2T = 0\)
- Equation for \(T\) in terms of basis functions gives approximate value
- Seek solution in which differential equation is satisfied in an average way over the region; \(w_i\) is weighting function

\[
\int_0^L w_i(x) \left[ \frac{d^2T}{dx^2} + a^2T \right] dx = 0 \quad i = 0, \ldots, N
\]
Modeling Differential Equation VI

- Integrals in summation are evaluated over each element from $x_i$ to $x_{i+1}$
- Substitute basis function equation for $T$ into these integrals
  \[ \widehat{T} = \sum_{j=0}^{N} T_j \phi_j \]
  \[ \int_{x_i}^{x_{i+1}} d\phi_i \frac{d}{dx} - \phi a^2 \phi_i \, dx = \sum_{j=0}^{N} T_j \int_{x_i}^{x_{i+1}} d\phi_j \frac{d}{dx} - \phi a^2 \phi_j \, dx \]
  \[ \int_{x_i}^{x_{i+1}} \phi a^2 \phi_i \, dx = \sum_{j=0}^{N} T_j \int_{x_i}^{x_{i+1}} \phi a^2 \phi_j \, dx \]

Results of Integration

- Linear Basis Functions
  \[ \phi_i(x) = \begin{cases} 
  0, & x \leq x_{i-1} \\
  \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x \leq x_i \\
  1, & x_i \leq x \leq x_{i+1} \\
  0, & x \geq x_{i+1} 
\end{cases} \]
  Get needed basis functions by substituting $i-1$ and $i+1$ for $i$
  Substitute basis functions and derivatives into integrals

- Constant Steps $x_{i+1} - x_i = h$
  \[ a_i = \frac{a^2}{3} (x_{i+1} - x_i) - \frac{1}{x_i - x_{i+1}} \]
  \[ b_i = \frac{a^2}{6} (x_{i+1} - x_i) + \frac{1}{x_i - x_{i+1}} \]
  \[ \sum_{j=0}^{N} T_j \int_{x_i}^{x_{i+1}} d\phi_i \frac{d}{dx} - \phi a^2 \phi_i \, dx = A_{i-1} T_{i-1} + A_i T_i + A_{i+1} T_{i+1} = 0 \]
  \[ A_{i-1} = \int_{x_i}^{x_{i+1}} d\phi_{i-1} \frac{d}{dx} - \phi a^2 \phi_{i-1} \, dx = b_{i-1} \]
  \[ A_i = \int_{x_i}^{x_{i+1}} d\phi_i \frac{d}{dx} - \phi a^2 \phi_i \, dx = a_i + a_{i-1} \]
  \[ A_{i+1} = \int_{x_i}^{x_{i+1}} d\phi_{i+1} \frac{d}{dx} - \phi a^2 \phi_{i+1} \, dx = b_i \]
Equations to be Solved

\[ \begin{align*}
\alpha_0 T_0 + \beta_0 T_1 &= \frac{dT}{dx} \bigg|_{x=0}, \\
\alpha_N T_{N-1} + \beta_N T_N &= \frac{dT}{dx} \bigg|_{x=s}
\end{align*} \]

\[ \beta_i T_{i+1} + (\alpha_i + \alpha_{i+1}) T_i + \beta_{i+1} T_{i+1} = 0 \quad i = 1, \ldots, N-1 \]

- Tridiagonal system of \( N+1 \) equations with \( N+3 \) variables
  - \( N+1 \) temperature values and 2 boundary gradients
  - Boundary conditions will specify two other equations

Boundary Gradients

- If we have Dirichlet boundary conditions, we can solve for temperatures then find gradients
- For Neumann or mixed boundary conditions, we must include gradients in tridiagonal solution
- Write boundary conditions as \( a \frac{dT}{dx} + bT = c \) and make \( g_0 = \frac{dT}{dx}|_{x=0} \) the first variable and \( g_L = \frac{dT}{dx}|_{x=L} \) the last one

Finite Element Equations

- Equations below only handle boundary conditions with specified gradients \( (a_0 \neq 0) \)

\[
\begin{bmatrix}
\alpha_0 & b_0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
-1 & \alpha_1 & \beta_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & \beta_0 & \alpha_0 + \alpha_1 & \beta_1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \beta_1 & \alpha_1 + \alpha_2 & \beta_2 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & \beta_2 & \cdots & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & \beta_{N-1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & \beta_{N-1} & \alpha_{N-1} + \alpha_0 & \beta_1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & \beta_N & \alpha_N + \alpha_1 & \beta_0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \beta_N & \alpha_N + \alpha_{N-1} & \beta_N & \beta_0 & c_0 & c_N
\end{bmatrix}
\]

Solution Errors for \( a = 2 \)

<table>
<thead>
<tr>
<th>N</th>
<th>100</th>
<th>100</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 RMS</td>
<td>1.7x10^{-5}</td>
<td>1.7x10^{-5}</td>
<td>1.8x10^{-3}</td>
<td>1.8x10^{-3}</td>
</tr>
<tr>
<td>( e_{\text{max}} )</td>
<td>2.4x10^{-5}</td>
<td>2.4x10^{-5}</td>
<td>2.4x10^{-3}</td>
<td>2.4x10^{-3}</td>
</tr>
<tr>
<td>( e_{\text{grad}(0)} )</td>
<td>3.6x10^{-4}</td>
<td>7.0x10^{-6}</td>
<td>7.0x10^{-2}</td>
<td>7.0x10^{-3}</td>
</tr>
<tr>
<td>( e_{\text{grad}(L)} )</td>
<td>2.1x10^{-4}</td>
<td>9.6x10^{-5}</td>
<td>1.8x10^{-2}</td>
<td>9.5x10^{-3}</td>
</tr>
</tbody>
</table>

Solution Errors for \( a = 0.2 \)

<table>
<thead>
<tr>
<th>N</th>
<th>100</th>
<th>100</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 RMS</td>
<td>6.2x10^{-10}</td>
<td>6.2x10^{-10}</td>
<td>6.5x10^{-8}</td>
<td>6.5x10^{-8}</td>
</tr>
<tr>
<td>( e_{\text{max}} )</td>
<td>8.6x10^{-10}</td>
<td>8.6x10^{-10}</td>
<td>8.5x10^{-8}</td>
<td>8.5x10^{-8}</td>
</tr>
<tr>
<td>( e_{\text{grad}(0)} )</td>
<td>1.3x10^{-6}</td>
<td>2.2x10^{-9}</td>
<td>1.3x10^{-4}</td>
<td>2.2x10^{-7}</td>
</tr>
<tr>
<td>( e_{\text{grad}(L)} )</td>
<td>1.3x10^{-6}</td>
<td>4.5x10^{-9}</td>
<td>1.3x10^{-4}</td>
<td>4.4x10^{-7}</td>
</tr>
</tbody>
</table>

Notes on the Error

- The formulations used here for finite elements and finite differences have second order error
  - Notes both equations almost the same
- Although temperature errors are similar, finite elements gives smaller errors in the gradients
- The heat source parameter, \( a^2 = b/k \), can change the error for a given \( h \)
Finite Element Grids

- Elements allow fitting complex objects used in almost all engineering designs
- Modern engineering software usually has grid generation that allows users to specify overall data on grid sizes and then has a program that generates the finite-element grid
- Element quality is a prime concern when considering the grid generated

Grid (Mesh) Quality

- Finite element mesh quality
- Grid generation programs for finite-element analysis of engineering problems report measures of grid quality
  - Skewness
  - Smoothness
  - Aspect ratio

Mesh Quality: Skewness

- Based on difference from an equilateral element
- Use quadrilateral elements as an example: equilateral elements have 90-degree angles
- Skewness = \( \frac{\theta_{\text{max}} - 90}{90 - \theta_{\text{min}}} \)
- Best value is zero; worst value is 1

Other Mesh Quality Issues

- Resolution – mesh should be finer in areas where there are significant changes such as fluid boundary layers, and stress concentrations
- Smoothness – changes in element sizes should be gradual
- Cell aspect ratios should usually not deviate more than 20% from uniform shaped cells except in special cases