Chapter 4

Introduction to finite element analysis

Introduction to finite element analysis

- FEA of the 1-D wave equation
- FEA of a rectangular thin plate
- Imposition of boundary conditions
- Examples

FEA of the 1-D wave equation

- Equation of motion
- Mesh and degrees of freedom
- Assembly of global matrices

FEA of the 1-D wave equation

- We first consider the discretization of the one-dimensional, unforced, linear wave equation using simple finite elements.
- For simplicity, we start with a dimensionless form of the equation on a domain of length 1.
- This governing partial differential equation is

$$u_{tt}(x,t)+\lambda u_t(x,t)-u_{xx}(x,t)=0, \quad \text{where} \quad u_x=\frac{\partial u}{\partial x}, \quad u_t=\frac{\partial u}{\partial t}, \ldots.$$

- We assume homogeneous boundary conditions

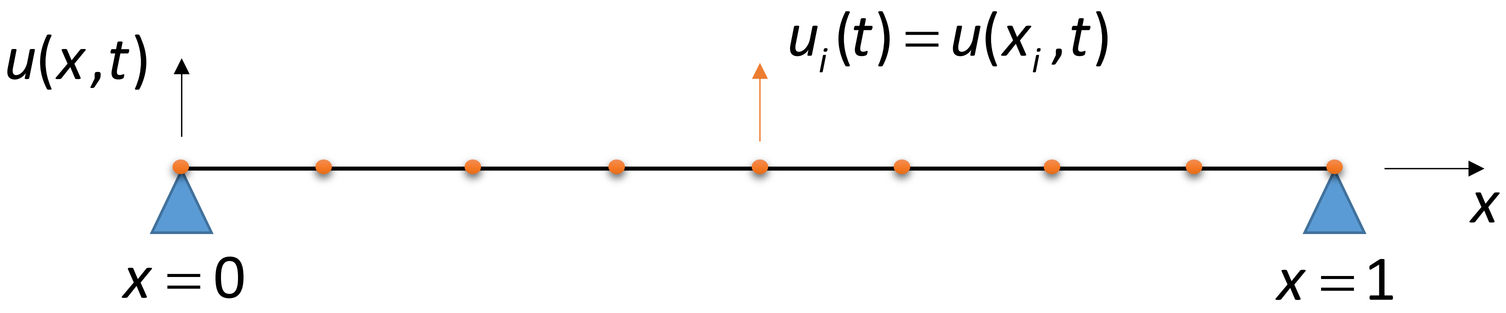
$$u(0,t)=0 \quad \text{or} \quad u_x(0,t)=0 \qquad \text{and} \qquad u(1,t)=0 \quad \text{or} \quad u_x(1,t)=0$$

and known initial conditions

$$u(x,0)=u_0(x) \qquad \text{and} \qquad u_t(x,0)=\dot{u}_0(x).$$

FEA of the 1-D wave equation

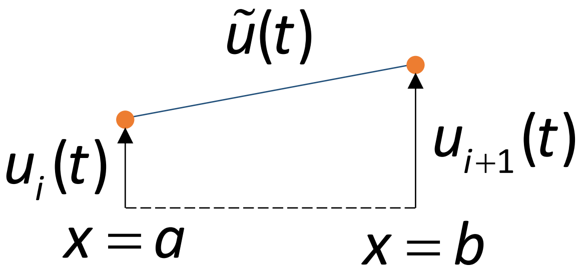
- We can obtain a discrete problem by dividing the problem domain into disjoint elements and interpolating an approximate solution within each element in terms of the solution at its endpoints, the element's nodes.



- **By discretizing the structure in space, we replace the PDE for $u(x,t)$ with a set of ODEs for $u_i(t), \quad i = 1, \ldots, N.$**
- We can then discretize time using the methods we've discussed for solving IVPs.

FEA of the 1-D wave equation

- The same PDE can give rise to different discrete models depending on how we choose to interpolate the approximate solution.
- We will make the simplest possible assumption, that the solution within an element depends linearly on the solution at its nodes. Let element *i* extend from $x_i = a$ to $x_{i+1} = b$.



- Considering a single element *i* extending from *x* = *a* to *x* = *b*, $0 \le a < b \le 1$, we can find the element mass, damping, and stiffness matrices.

$$\mathbf{m}_i = \frac{b-a}{2}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{c}_i = \lambda \frac{b-a}{2}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{k}_i = \frac{1}{b-a}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

FEA of the 1-D wave equation

- The discrete equation of motion for the entire domain is

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0},$$

  where the global matrices **M**, **C**, and **K** have been assembled from the element matrices **m**, **c**, and **k** (which may differ from element to element) and where **u**(*t*) is a vector of all the nodal displacements (unknowns in the solution).
- The assembly of the global matrices and the imposition of the BCs is "just bookkeeping," but it's easy to get it wrong in a program.
- We need to systematically associate element degrees of freedom (one per node here) and global degrees of freedom (ranging from 1 to *N*, with *N* determined by how we define the mesh).
- Many texts make this harder than it needs to be.

FEA of the 1-D wave equation

- We'll take the assembly of the global mass matrix as an example.
- For each element *i*, we determine the global DOF *j* and *k* corresponding to the element's local DOF 1 and 2.
  - Because of how we've numbered the nodes in our 1-D mesh, $j = i$ and $k = i + 1$.
  - **More generally,** we'll need to store these and look them up for each element.
- We start with an *N* by *N* matrix **M** of all zeros.
- For each element *i*, we add the scalar elements of **m** to **M**.

$$M_{jj} = M_{jj} + m_{11}, \quad M_{jk} = M_{jk} + m_{12}, \quad M_{kj} = M_{kj} + m_{21}, \quad M_{kk} = M_{kk} + m_{22}$$

- The result is an *N* by *N* matrix, which we can expect to be symmetric.

FEA of the 1-D wave equation

- We can repeat exactly the same steps for the damping and stiffness matrices.
- We end up with the global **M**, **C**, and **K** matrices, which are the coefficients of an equation of the form

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0}.$$

- We can't solve this yet because we haven't taken account of the boundary conditions.
    - We will defer this until we study the FE formulation for another structure.

FEA of a rectangular thin plate

Relax, I'm not going to assign it this term. But we should discuss

- Equation of motion
- Mesh and degrees of freedom
- Assembly of global matrices

FEA of a rectangular thin plate

- For us, most of the complexity of the thin plate is due to the second spatial dimension.
- Its governing equation is

$$\rho h \ddot{w}(x,y,t) + C\dot{w}(x,y,t) + D\nabla^4 w(x,y,t) = 0$$

where ρ is the mass density (per unit volume) of the plate, h is its thickness, C is the viscous damping coefficient (per unit area),

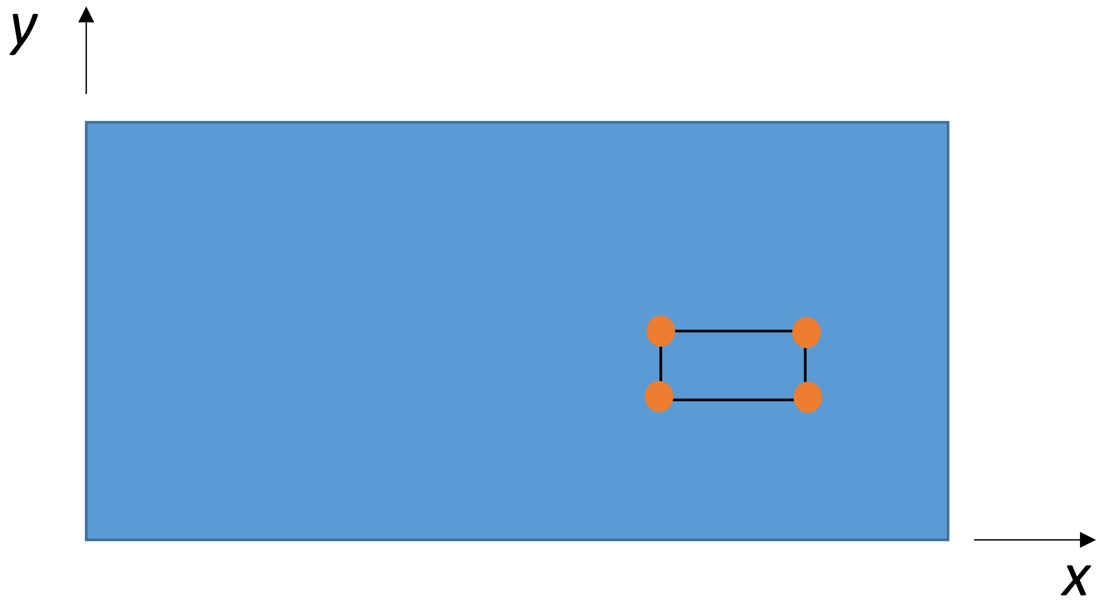
$$D = \frac{Eh^3}{12(1-\nu^2)}, \quad \nabla^4 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4},$$

and E is Young's modulus, and ν is Poisson's ratio.

- We will also assume that some points (typically on the boundaries) are fixed.

FEA of a rectangular thin plate

- We will assume a right-handed *xyz* coordinate system, with its origin at one corner of the rectangular plate.



- We've indicated a typical element of the plate domain, defined by 4 nodes.

FEA of a rectangular thin plate

- Recall that for the string, each node had 1 DOF, representing transverse displacement. Consequently, each element coupled two degrees of freedom of the global problem.
- In the plate element, each node typically has 3 DOF:
  - Transverse displacement in the *z* direction, $w_i = w(x_i, y_i, t)$.
  - Rotation about the *x* axis, $\theta_i = \theta(x_i, y_i, t)$.
  - Rotation about the *y* axis, $\phi_i = \phi(x_i, y_i, t)$.
  - Other choices are possible!
- Therefore,
  - Each element mass, damping, and stiffness matrix is 12 x 12.
  - A mesh with *N* nodes has 3*N* DOF (neglecting boundary conditions).

FEA of a rectangular thin plate

- Assuming a total of N nodes, we can define the global displacement vector as

$$\mathbf{u} = \begin{bmatrix} w_1 & \theta_1 & \phi_1 & w_2 & \theta_2 & \phi_2 & \cdots & w_N & \theta_N & \phi_N \end{bmatrix}^T .$$

- This implicitly determines which 12 global DOF are coupled by each element matrix.
- Other than the number of rows and columns involved (12 rather than 2), the assembly of the global matrices is exactly the same as for the string. As we said before, it's bookkeeping…and errors are easy to make.
- The precise form of the element matrices depends on the details of the assumed interpolating functions (and several more subtle effects).

FEA of a rectangular thin plate

- We define the element degrees of freedom in the order

$$\bar{\mathbf{u}} = \begin{bmatrix} \bar{w}_1 & \bar{\theta}_1 & \bar{\phi}_1 & \bar{w}_2 & \bar{\theta}_2 & \bar{\phi}_2 & \bar{w}_3 & \bar{\theta}_3 & \bar{\phi}_3 & \bar{w}_4 & \bar{\theta}_4 & \bar{\phi}_4 \end{bmatrix}^T$$

  where the subscripts correspond to the nodes of the element, (usually) numbered counterclockwise (i.e., the element local coordinate system is also right-handed).

- The various nodal transverse displacements and rotations, denoted by overbars, are aligned with the global displacements and rotations.
- Assembling the global matrices requires mapping the 12 element DOF to 12 of the N global DOF, then looping over the elements and doing 144 scalar additions to the global matrix for each element matrix.
  - That sounds much worse than it is. But don't do it manually.

FEA of a rectangular thin plate

- We can exploit symmetry in writing out the element matrices.
  - These will be given in a separate handout.

Imposition of boundary conditions

- To impose the boundary conditions, we begin by identifying the unconstrained and constrained DOF. The former correspond to the solution of the problem, while the latter are typically prescribed displacements (often zero).
- We then partition the global matrix equation, rearranging rows and columns as necessary, to produce

$$\begin{bmatrix} \mathbf{M}_{uu} & \mathbf{M}_{uc} \\ \mathbf{M}_{cu} & \mathbf{M}_{cc} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_u \\ \ddot{\mathbf{u}}_c \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{uu} & \mathbf{C}_{uc} \\ \mathbf{C}_{cu} & \mathbf{C}_{cc} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_u \\ \dot{\mathbf{u}}_c \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uc} \\ \mathbf{K}_{cu} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_u \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

  where the subscripts *u* and *c* denote unknown and known (constrained) displacements, respectively.
- We can move the known displacements (boundary conditions) to the right-hand side.

Imposition of boundary conditions

- The result is

$$\mathbf{M}_{uu}\ddot{\mathbf{u}}_u + \mathbf{C}_{uu}\dot{\mathbf{u}}_u + \mathbf{K}_{uu}\mathbf{u}_u = -\mathbf{M}_{uc}\ddot{\mathbf{u}}_c - \mathbf{C}_{uc}\dot{\mathbf{u}}_c - \mathbf{K}_{uc}\mathbf{u}_c\,.$$

- If the BCs are homogeneous (steady, zero displacements), the RHS vanishes.
- If the BCs are prescribed motions, the RHS is nonzero, but it is known.