The Finite Element Method

Section 9

Dr. Matthew Santer

Department of Aeronautics, I.C.L.

m.santer@imperial.ac.uk

Section Objectives

To complete our introduction to the finite element method we now extend our analysis of 2D boundary value problems to include those for which the variable is a vector ($\mathbf{u}$). The example we will use as a framework for the developed techniques is classical linear elastostatics (which is mathematically identical to Stokes Flow).

In this section we will:

- Use the finite element method to approximate a 2-D BVP with a vector field
- Illustrate (for the final time) the $(S) \to (W) \to (G) \to (M)$ process to produce a finite-element approximation

Mathematical Preliminaries

In order to get a weak form of the 2-D BVP with vector field it is necessary to introduce the Euclidean decomposition of a general (non-symmetric) 2nd-rank tensor s_{ij} into symmetric and skew-symmetric components

$$s_{ij} = s_{(ij)} + s_{[ij]}$$

in which the symmetric part obeys the relationship $s_{(ij)} = s_{(ji)}$, and the skew-symmetric part obeys the relationship $s_{[ij]} = -s_{[ji]}$. These symmetric and skew-symmetric components are defined as

$$s_{(ij)} \overset{def}{=} \frac{s_{ij} + s_{ji}}{2} \quad \text{and} \quad s_{[ij]} \overset{def}{=} \frac{s_{ij} - s_{ji}}{2}$$

We now consider the multiplication of a non-symmetric tensor s_{ij} and a symmetric tensor t_{ij}

Visualizer

Demonstrate that $s_{ij}t_{ij} = s_{(ij)}t_{ij}$

Classical Linear Elastostatics

The physical problem which we will use as the framework for solving a BVP with a vector variable is classical linear elastostatics. The analysis is valid for both two and three spatial dimensions, although we will focus exclusively on the former. We take dummy variables

$$i, j, k = 1, \cdots, n_{sd}$$

where n_{sd} is the number of spatial dimensions. We then define

σ_{ij}	(Cartesian components of the Cauchy stress tensor)
u_i	(displacement vector)
l_i	(prescribed body force per unit volume)
ϵ_{ij}	(infinitesimal strain tensor)

The infinitesimal strain tensor is defined as the symmetric component of the gradient of the displacements

$$\epsilon_{ij} = u_{(i,j)} \stackrel{def}{=} \frac{u_{i,j} + u_{j,i}}{2}$$

Classical Linear Elastostatics: Strong Form

The stress and strain tensors are related by a constitutive equation – in this case Hooke's Law – according to

$$\sigma_{ij} = c_{ijkl}\epsilon_{kl}$$

in which c_{ijkl} are the elastic coefficients, which are themselves functions of $\mathbf{x}$ if the continuum material is not homogeneous. It is a property of the elastic coefficient tensor that it is symmetric, i.e. $c_{ijkl} = c_{klij}$.

The strong form of the classical linear elastostatics problem is

$$(\mathrm{S}) \qquad \left\{ \begin{array}{rcll} \sigma_{ij,j} + l_i & = & 0 & \text{in } \Omega \qquad \text{(Equilibrium)} \\ u_i & = & g_i & \text{on } \Gamma_{g_i} \quad \text{(prescribed boundary displacement)} \\ \sigma_{ij} n_j & = & h_i & \text{on } \Gamma_{h_i} \quad \text{(prescribed boundary traction)} \end{array} \right.$$

where σ_{ij} is defined in terms of the variable u_i by

$$\sigma_{ij} = c_{ijkl}\left(\frac{u_{k,l} + u_{l,k}}{2}\right)$$

Classical Linear Elastostatics: Weak Form

We obtain the weak form of the problem in a similar fashion to previous examples: we require the integral of the weighted residual over the domain to be equal to zero, and we integrate by parts to result in symmetric functions.

Visualizer

Show that the weak form of the classical linear elastostatics problem is, for

$$u_i = g_i \quad \text{on } \Gamma_{g_i}$$

$$\text{(W)} \qquad a(\mathbf{w}, \mathbf{u}) = (\mathbf{w}, \mathbf{l}) + (\mathbf{w}, \mathbf{h})_\Gamma$$

where

$$a(\mathbf{w}, \mathbf{u}) = \int_\Omega w_{(i,j)} c_{ijkl} u_{(k,l)}, \quad (\mathbf{w}, \mathbf{l}) = \int_\Omega w_i l_i d\Omega, \quad (\mathbf{w}, \mathbf{h})_\Gamma = \sum_{i=1}^{n_{sd}} \left(\int_{\Gamma_{h_i}} w_i h_i d\Gamma \right)$$

Matrix/Vector Notation

If subscript notation doesn't agree with you, the derived symmetric bilinear functions in the weak form may be expressed in matrix/vector notation

Visualizer

Show the symmetric bilinear function $a(\mathbf{w}, \mathbf{u})$ can be expressed as

$$a(\mathbf{w}, \mathbf{u}) = \int_{\Omega} \epsilon(\mathbf{w})^T \mathbf{D} \epsilon(\mathbf{u}) d\Omega$$

Galerkin Form

We proceed as before, accounting for the vectorial nature of the variables. $\mathcal{S}^h$ and $\mathcal{V}^h$ are finite-dimensional approximations to the trial solution space $\mathcal{S}$ and the variation space $\mathcal{V}$ respectively.

We require that $\mathbf{w}^h \in \mathcal{V}^h$ (approximately) satisfies $w_i = 0$ on Γ_{g_i}.

We also require that members of $\mathcal{S}^h$ can be decomposed to $\mathbf{u}^h = \mathbf{v}^h + \mathbf{g}^h$, where $\mathbf{v}^h \in \mathcal{V}^h$, and $\mathbf{g}^h$ approximately satisfies $u_i = g_i$ on Γ_{g_i}.

The Galerkin form of the problem may therefore be expressed as

$$\text{(G)} \qquad a(\mathbf{w}^h, \mathbf{v}^h) = (\mathbf{w}^h, \mathbf{l}) + (\mathbf{w}^h, \mathbf{h})_\Gamma - a(\mathbf{w}^h, \mathbf{g}^h)$$

Approximation Functions

We first redefine the ID array to include the fact that each node has multiple degrees of freedom i.

$$ID(i, A) = \begin{cases} P & \text{if} \quad A \in \eta - \eta_{g_i} \\ 0 & \text{if} \quad A \in \eta_{g_i} \end{cases}$$

where A is the global node number and P is the global equation number.

We make the following approximations

$$v_i^h = \sum_{A \in \eta - \eta_{g_i}} N_A d_{iA}$$

$$g_i^h = \sum_{A \in \eta_{g_i}} N_A g_{iA}$$

Approximation Functions continued

We can express these components of the approximate solution in vector notation with reference to the Euclidean vector basis, i.e. in 2 dimensions

$$\mathbf{e}_1 = \left\{ \begin{array}{c} 1 \\ 0 \end{array} \right\} \quad \mathbf{e}_2 = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}$$

giving

$$\mathbf{v}^h = v_i^h \mathbf{e}_i \qquad \mathbf{g}^h = g_i^h \mathbf{e}_i$$

and likewise

$$\mathbf{w}^h = w_i^h \mathbf{e}_i \qquad \text{where} \qquad w_i^h = \sum_{A \in \eta - \eta_{g_i}} N_A c_{iA}$$

Galerkin Form

We then substitute these approximations into the Galerkin form of the problem (G) to obtain

$$a\left(\sum_{A\in\eta-\eta_{g_i}} N_A c_{iA}\mathbf{e}_i \ , \ \sum_{j=1}^{n_{dof}}\left(\sum_{B\in\eta-\eta_{g_i}} N_B d_{jB}\mathbf{e}_j\right)\right) =$$

$$\left(\sum_{A\in\eta-\eta_{g_i}} N_A c_{iA}\mathbf{e}_i \ , \ \mathbf{l}\right) + \left(\sum_{A\in\eta-\eta_{g_i}} N_A c_{iA}\mathbf{e}_i \ , \ \mathbf{h}\right)_\Gamma$$

$$-a\left(\sum_{A\in\eta-\eta_{g_i}} N_A c_{iA}\mathbf{e}_i \ , \ \sum_{j=1}^{n_{dof}}\left(\sum_{B\in\eta-\eta_{g_i}} N_B g_{jB}\mathbf{e}_j\right)\right)$$

Towards the Matrix Form

By invoking the symmetry and bilinearity of the chosen functions, this may be re-expressed as

$$\text{for } A \in \eta - \eta_{g_i} \qquad \text{and} \qquad 1 \le i \le n_{sd}$$

$$\sum_{j=1}^{n_{dof}} \left(\sum_{B \in \eta - \eta_{g_i}} a(N_A \mathbf{e}_i \ , \ N_B \mathbf{e}_j) d_{jB} \right) = $$

$$(N_A \mathbf{e}_i, \mathbf{l}) + (N_A \mathbf{e}_i, \mathbf{h})_\Gamma - \sum_{j=1}^{n_{dof}} \left(\sum_{B \in \eta_{g_j}} a(N_A \mathbf{e}_i \ , \ N_B \mathbf{e}_j) g_{jB} \right)$$

Matrix Form

This equation may be concisely expressed as

$$\mathbf{Kd} = \mathbf{F}$$

or

$$[K_{PQ}]\{d_Q\} = \{F_P\}$$

where

$$\begin{aligned}
K_{PQ} &= a(N_A\mathbf{e}_i, N_B\mathbf{e}_j) \\
F_P &= (N_A\mathbf{e}_i, \mathbf{l}) + (N_A\mathbf{e}_i, \mathbf{h})_\Gamma - \sum_{j=1}^{n_{dof}} \left(\sum_{B\in\eta_{g_j}} a(N_A\mathbf{e}_i, N_B\mathbf{e}_j) g_{jB} \right) \\
P &= ID(i, A) \\
Q &= ID(j, B)
\end{aligned}$$

Summary

We will not cover it in this course, but a similar assembly operator algorithm to the scalar variable case may readily be derived.
In this section we have:

- Demonstrated the finite element approximation for a 2-D BVP with vector field variable; specifically the classical linear elastostatics problem
- Completed our introduction to the finite element method from mathematical first principles
- You can now, for problems with up to three spatial dimensions:
  - Derive an appropriate finite element approximation
  - Discretize the global domain into repeating locally-defined finite elements
  - Assemble global matrices and vectors from these local finite elements via mapping and an appropriate algorithm
  - Account time for time dependence if desired