The Finite Element Method

Section 2

Dr. Matthew Santer

Department of Aeronautics, I.C.L.

m.santer@imperial.ac.uk

Section Objectives

In the last section we introduced the Galerkin method as a refinement of the weighted residual technique for the solution of linear systems.

In this section we will:

- apply the Galerkin method to solve a simple 1-D problem
- first introduce the (S) $\rightarrow$ (W) $\rightarrow$ (G) $\rightarrow$ (M) procedure
- introduce finite elements in a globally-defined form

Preliminaries

We will use subscript notation for conciseness. The form we will adopt uses subscripts to indicate differentiation.

A subscript comma indicates differentiation with respect to the variable following the comma. If there are n variables following the comma, this indicates the nth differential. For example

$$u_{,xx} = \frac{d^2 u}{dx^2}$$

Sobolev Space

$u \in H^1 \implies u_{,x}$ is square-integrable, i.e.

$$\int_0^1 (u_{,x})^2 dx < \infty$$

1-D Poisson Equation

Consider the solution of the one-dimensional Poisson equation over the domain $\Omega = [0, 1]$ with both a Dirichlet and a Neumann boundary condition.

This could represent, for example, a 1-D Heat conduction problem.

Our problem is a two-point BVP.

$$(S) \begin{cases} u_{,xx} + l &= 0 \quad \text{on } \Omega \\ u(1) &= g \quad \text{(Dirichlet)} \\ -u_{,x}(0) &= h \quad \text{(Neumann)} \end{cases}$$

(S) indicates the strong form of the problem

The Weak Form

The first step in the process is to convert (S) to the weak form. We will use the set of trial solutions $\mathcal{S}$ and the set of weight function $\mathcal{V}$

$$\begin{aligned} \mathcal{S} &= \{u | u \in H^1, u(1) = g\} \\ \mathcal{V} &= \{w | w \in H^1, w(1) = 0\} \end{aligned}$$

Visualizer

Convert the strong form (S) to the weak form (W) where

$$(W) \quad a(w, u) = (w, l) + w(0)h$$

The Galerkin Method I

We now consider a discretization (or mesh) of the domain Ω represented by a superscript h. This gives us the discrete trial and weight functions as

$$\mathcal{S}^h \subset \mathcal{S} \quad \text{and} \quad \mathcal{V}^h \subset \mathcal{V}$$

which implies that if $u^h \in \mathcal{S}^h$ then $u^h \in \mathcal{S}$, and similarly if $w^h \in \mathcal{V}^h$ then $w^h \in \mathcal{V}$. Also we require $u^h(1) = g$ and $w^h(1) = 0$.

We assume that $\mathcal{V}^h$ is given and has individual members v^h, i.e. $v^h \in \mathcal{V}^h$

The members of the discrete solution set are

$$u^h = v^h + g^h, u^h \in \mathcal{S}^h$$

in which g^h is a given function satisfying the essential boundary condition, i.e. $g^h(1) = g$, and hence $u^h(1) = v^h(1) + g^h(1) = g$.

The Galerkin Method II

We can substitute these discrete solutions into the symmetric bilinear form of the weak expression (W)

$$a(w^h, u^h) = (w^h, l) + w^h(0)h$$

Expanding u^h into its constitute parts and rearranging gives the Galerkin form of the problem. In full this is:

for a given l, g, and h, find $u^h = v^h + g^h, v^h \in \mathcal{V}^h$ subject to $\forall\ w^h \in \mathcal{V}^h$

$$(G) \quad a(w^h, v^h) = (w^h, l) + w^h(0)h - a(w^h, g^h)$$

(G) indicates the Galerkin form of the problem

Shape Functions

We now need to say something about the form of the discrete weighting functions. We specify

$$w^h \ (\in \mathcal{V}^h) = \sum_{A=1}^{n} c_A N_A = c_1 N_1 + c_2 N_2 + \cdots + c_n N_n$$

where N_A are the 'shape functions'.

$N_A(1) = 0$ in order to satisfy the requirement that $w^h(1) = 0$. One additional shape function N_{n+1} is needed which has the property $N_{n+1}(1) = 1$. This enables us to write $g^h = gN_{n+1}$ and hence $g^h(1) = 1$ as required.

We then use the same shape functions to represent the solution

$$u^h = v^h + g^h = \sum_{A=1}^{n} d_A N_A + gN_{n+1}$$

Towards the matrix form ...

We can now substitute our expressions for u^h and w^h into G to get

$$a\left(\sum_{A=1}^{n} c_A N_A, \sum_{B=1}^{n} d_B N_B\right) = \left(\sum_{A=1}^{n} c_A N_A, l\right) + \left(\sum_{A=1}^{n} c_A N_A(0)\right) h - a\left(\sum_{A=1}^{n} c_A N_A, g N_{n+1}\right)$$

We can now invoke the bilinearity of $a(\cdot,\cdot)$ and $(\cdot,\cdot)$ to re-express this as

$$0 = \sum_{A=1}^{n} c_A \left(\sum_{B=1}^{n} a(N_A, N_B) d_B - (N_A, l) - N_A(0) h + a(N_A, N_{n+1}) g\right)$$

or, more consisely

$$0 = \sum_{A=1}^{n} c_A G_A$$

The matrix form

As c_A is nonzero and arbitrary $\implies$ $G_A = 0$ hence

$$\sum_{B=1}^{n} a(N_A, N_B)d_B = (N_A, l) + N_A(0)h - a(N_A, N_{n+1})g$$

We now define

$$\begin{aligned} K_{AB} &= a(N_A, N_B) \\ F_A &= (N_A, l) + N_A(0)h - a(N_A, N_{n+1})g \end{aligned}$$

which enables us to write

$$\sum_{B=1}^{n} K_{AB} d_B = F_A \quad A, B = 1, 2, \cdots, n$$

The matrix form

In matrix notation this can be expressed as

$$(M) \quad [K_{AB}]\{d_B\} = \{F_A\} \quad \text{or} \quad \mathbf{Kd} = \mathbf{F}$$

(M) indicates the Matrix form of the problem

Note we have followed the path $(S) \rightarrow (W) \rightarrow (G) \rightarrow (M)$. This will still be the case as we increase the complexity of the problems we are solving.

Symmetry of $a(\cdot, \cdot)$ means that $K_{AB} = K_{BA}$ or $\mathbf{K} = \mathbf{K}^T$ which has important computational ramifications. This is the advantage of our specific choice of (W).

Summary

- We have used the Galerkin method to reduce a 1-D BVP to an approximate matrix form that is easily solved
- We have followed the path $(S) \rightarrow (W) \rightarrow (G) \rightarrow (M)$. This will be the case for all the problems we consider in this course
- The accuracy of the approximate solution is dependent on the choice of shape functions
- Is a shape function a finite element?

Visualizer

Example of a two d.o.f. approximation, i.e.

$$n = 2$$