Finite Element Approximations

Applied Mathematics, Studio Session 1

1 Laplace Equation

Today, we are going to solve the *Laplace* equation

$$\begin{aligned} -u'' = 0, &\quad 0 < x < 1, \\ u(0) = 0, &\quad u'(1) = g_1, \end{aligned} \tag{1}$$

where g_1 is a constant.

Our basic solution tool will be the *Finite Element Method.*

2 Variational Formulation

A *Finite Element Method* is usually derived by using a variational formulation of the problem at hand.

Multiply $-u'' = 0$ by a so-called test function v satisfying $v(0) = 0$ and integrate, i.e.,

$$\int_0^1 -u''v\,dx = 0. \tag{2}$$

If v is smooth we can integrate by parts to get

$$\begin{aligned} -\int_0^1 u''v\,dx &= -\Big[u'(x)v(x)\Big]_0^1 + \int_0^1 u'v'\,dx \\ &= -u'(1)v(1) + u'(0)v(0) + \int_0^1 u'v'\,dx = 0. \end{aligned} \tag{3}$$

Since $u'(1) = g_1$ and $v(0) = 0$ we conclude that u satisfies

$$\int_0^1 u'v'\,dx = g_1 v(1), \qquad \forall v \in V, \tag{4}$$

where

$$V = \left\{ v(x) : \int_0^1 |v'(x)|^2 + |v(x)|^2 \, dx < \infty, \; v(0) = 0 \right\}.$$

Question 1. Give a few examples of test functions.

3 Finite Element Method

As you might have guessed there are infinitely many test functions v and the space V containing them is of infinite dimension.

Suppose now that U is an approximation to the solution u for (1). Obviously U does not satisfy (4) for every test function $v \in V$ since only u has this property. However, if we relax the assumption that the variational formulation must hold for all $v \in V$, i.e., restrict the number of test functions to N say, then it is possible to construct U so that (4) is satisfied for at least these N test functions.

Ideas such as those above lies at the very hart of the *Finite Element Method.* It reads: find $U \in V_h$ such that

$$\int_0^1 U'v' \, dx = g_1 v(1), \qquad \text{forall} \quad v \in V_h, \tag{5}$$

where V_h is the function space containing the reduced (finite) set of test functions, i.e., V_h is finite dimensional. To construct a set of basis functions for V_h, we start with a partition

$$\mathcal{T}_h : 0 = x_0 < x_1 < x_2 < \ldots < x_N = 1, \tag{6}$$

of the interval $0 \leq x \leq 1$ into N subintervals of equal length h.

Here, we choose V_h as the space of continuous piecewise linear functions on the partition $\mathcal{T}_h$, vanishing at the left end-point $x = 0$ (since $u(0) = 0$). For this choice of V_h, the set $\{\varphi_i\}_1^N$ associated with the $N + 1$ nodes x_j, $j = 0, 1, \ldots, N$, such that

$$\varphi_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases} \tag{7}$$

forms a set of basis functions for V_h. Note that we have NOT! defined a base function φ_0 corresponding to $x = 0$ (with $\varphi_0(0) = 1$). This is due to the fact that, since $u(x_0) = u(0) = 0$, all the functions of V_h must vanish at $x = 0$. Therefore, here, we cannot have a base function φ_0.

We now seek to solve (5) using the continuous piecewise linear ansatz

$$U(x) = \xi_1\varphi_1(x) + \xi_2\varphi_2(x) + \ldots \xi_N\varphi_N(x) = \sum_{i=1}^{N} \xi_i\varphi_i(x), \tag{8}$$

where ξ_i, $i = 1, 2, \ldots, N$ are N coefficients to be determined such that (5) is satisfied. This is accomplished by evaluating (5) for N different choises of test functions v. As a result we get a system of equations from which the coefficients ξ_i can be found. One way of choosing these test functions is to set $v = \varphi_j$, $j = 1, \ldots, N$, because if (5) is satisfied for each basis function φ_j separately, then it is also satisfied for all $v \in V_h$.

Substituting $U = \xi_1\varphi_1 + \xi_2\varphi_2 + \ldots \xi_N\varphi_N$ into (5) and choosing $v = \varphi_j$, $j = 1, 2, \ldots, N$, we get

$$\left\{ \begin{array}{c} \displaystyle\int_0^1 (\xi_1\varphi_1' + \xi_2\varphi_2' + \ldots + \xi_N\varphi_N')\varphi_1'\, dx = g_1\varphi_1(1) \\ \displaystyle\int_0^1 (\xi_1\varphi_1' + \xi_2\varphi_2' + \ldots + \xi_N\varphi_N')\varphi_2'\, dx = g_1\varphi_2(1) \\ \vdots \\ \displaystyle\int_0^1 (\xi_1\varphi_1' + \xi_2\varphi_2' + \ldots + \xi_N\varphi_N')\varphi_N'\, dx = g_1\varphi_N(1) \end{array} \right. , \tag{9}$$

which is a $(N \times N)$ linear system of algebraic equations

$$\boldsymbol{A\xi} = \boldsymbol{b}, \tag{10}$$

where the entries of the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ are given by

$$A_{ij} = \int_0^1 \varphi_i'(x)\, \varphi_j'(x)\, dx, \qquad b_j = g_1\varphi_j(1), \qquad i, j = 1, 2, \ldots, N. \tag{11}$$

Here, ξ denote a vector holding the coefficients ξ_i.

Question 2. Derive (10) from (9).

4 Computer Assignment

Write a finite element computer code that solves (1) using *MATLAB*.

- Start by assuming $N = 3$ and calculate by hand the entries of $\boldsymbol{A}$ and $\boldsymbol{b}$ if V_h is spanned by the bases

$$\varphi_1(x) = \begin{cases} 3x, & 0 < x < 1/3 \\ 2 - 3x, & 1/3 < x < 2/3 \\ 0, & 2/3 < x < 1 \end{cases}, \tag{12}$$

$$\varphi_2(x) = \begin{cases} 0, & 0 < x < 1/3 \\ 3x - 1, & 1/3 < x < 2/3 \\ 3 - 3x, & 2/3 < x < 1 \end{cases}, \tag{13}$$

$$\varphi_3(x) = \begin{cases} 0, & 0 < x < 1/3 \\ 0, & 1/3 < x < 2/3 \\ 3x - 2, & 2/3 < x < 1 \end{cases}. \tag{14}$$

  Solve the system $\boldsymbol{A}\boldsymbol{\xi} = \boldsymbol{b}$ numerically and plot U.

- Improve the solution approximation U by increasing the number of intervals N.

  Use that if $\mathcal{T}_h : x_j = jh,\ j = 0, 1, \ldots, N$ is a uniform partition of $0 \leq x \leq 1$ into N subintervals of length h, then the basis functions are given by

$$\varphi_j(x) = \frac{1}{h} \begin{cases} x - x_{j-1}, & x_{j-1} < x < x_j \\ x_{j+1} - x, & x_j < x < x_{j+1} \\ 0, & \text{otherwise} \end{cases}, \quad j = 1, 2, \ldots, N, \tag{15}$$

  except for $\varphi_N(x)$ which vanish outside $x_{N-1} \leq x \leq x_N$.

- Extend the code to also solve the *Poisson* equation

$$\begin{aligned} -u'' &= f, \quad 0 < x < 1, \\ u(0) &= 0, \quad u'(1) = g_1, \end{aligned} \tag{16}$$

  where f is a constant. Start by deriving a new variational formulation.

5 Code Template

Below is a small *MATLAB* code template which might give you a few hints concerning program structure, for example.

```
N = 3 % three subintervals

h = 1/N % interval size

x = 0:h:1 % node points

A = zeros(N,N) % initialize stiffness matrix to zero

b = zeros(N,1) % initialize load vector to zero


% Compute A and b %


xi = A\b % solve system of eqs.

U = zeros(N+1,1) % initialize load vector

U(2:N+1) = xi % adjust for boundary condition u(0) = 0.

plot(x,U) % graph solution approximation
```