MAST90026 Computational Differential Equations: Week 3

Jesse Collis

Modified from Hailong Guo (2022)

Semester 1 2024

The University of Melbourne

Galerkin method

Consider the second order self-adjoint BVP:

$$-(D(x)u')' + q(x)u = f(x),$$

subject to Dirichlet boundary conditions $u(a) = \alpha$, $u(b) = \beta$. We also require $D(x) > 0, q(x) \geq 0$.

Idea of Galerkin mehtod: require the projection of the residual onto V to vanish, i.e. require

$$\int_a^b [-(D(x)u')' + q(x)u - f(x)]v(x)\,dx = 0, \qquad \forall v \in V,$$

where V is called the *test space*.

Integration by part

Idea: integrate by parts (if possible) and obtain the equation:

$$[-D(x)u'v]_a^b + \int_a^b D(x)u'v'\,dx + \int_a^b q(x)uv\,dx = \int_a^b fv\,dx, \quad \forall v \in V.$$

where $D(x)u'(x)$ is the flux.

Question: How to choose the test space V?

Answer: Depends on the boundary condition.

- Neumann/Robin boundary conditions, the flux or u' is known on the boundaries.
- Dirichlet boundary conditions, do not know the flux on the boundary. Remove it by requiring $v = 0$ on the boundaries. The "don't test where you know u" principle

Motivation for weak derivative

If a function f(x) is differentiable on $[a, b]$, then after integration by parts, for any smooth function $v(x)$ satisfying $v(a) = v(b) = 0$, we have

$$\int_a^b f(x)v'(x)dx = -\int_a^b f'(x)v(x)dx.$$

Example. The function $f(x) = |x|$ is not differentiable but we can define its weak derivative as the step function $g(x) = \begin{cases} 1, & x \geq 0, \\ -1, & x < 0. \end{cases}$

Weak derivative

Definition : (Weak derivative)

A function $g(x)$ is defined to be the weak derivative of f on $[a, b]$ if it satisfies

$$\int_a^b f(x)v'(x)dx = -\int_a^b g(x)v(x)dx, \quad \forall v(x) \in C_0^\infty([a, b]),$$

where $C^\infty([a, b])$ is the set of functions with derivatives of any order being continuous and $C_0^\infty([a, b])$ denotes that functions should also vanish at the boundaries $x = a$ and $x = b$.

Denote $f'(x) = g(x)$.

Weak derivative (continue)

Definition : (kth order weak derivative)

A function $g(x)$ is defined to be the kth order weak derivative of f on $[a, b]$ if it satisfies

$$\int_a^b f(x)v^{(k)}(x)dx = (-1)^k \int_a^b g(x)v(x)dx, \quad \forall v(x) \in C_0^\infty([a, b]).$$

Sobolev space

Definition : (Sobolev space)

For a positive index k, the Sobolev space $H^k(a,b)$ is the set of functions $u:(a,b)\to\mathbb{R}$ such that u and all (weak-) derivatives up to and including k are square integrable:

$$u \in H^k(a,b) \Longleftrightarrow \int_a^b u^2 < \infty, \int_a^b (u')^2 < \infty, \dots, \int_a^b \left(u^{(k)}\right)^2 < \infty$$

Note that $H^k(a,b)$ defines a Hilbert space with inner product

$$(u,w)_k = \int_a^b uw + \int_a^b u'w' + \int_a^b u''w'' + \ldots + \int_a^b u^{(k)}w^{(k)},$$

and norm

$$\|u\|_k = \left(\int_a^b u^2 + \int_a^b (u')^2 + \ldots + \int_a^b \left(u^{(k)}\right)^2\right)^{1/2}.$$

Returning to model equation

The appropriate space for V turns out to be

$$H_0^1(a,b) = \underbrace{\{u \in L_2(a,b), u' \in L_2(a,b)}_{u \in H^1(a,b)}; \underbrace{u(a) = 0, u(b) = 0\}}_{\text{essential b.c.'s}}.$$

This is a big surprise:

- There are no second derivatives in the definition of H_0^1 !
- First derivatives need not be continuous!

Weak form

Definition : (Variational form)

The variational form of the model equation is to find $u \in H^1(a,b)$ with $u(a) = \alpha$ and $u(b) = \beta$ such that

$$\int_a^b D(x) u' v' \, dx + \int_a^b q(x) u v \, dx = \int_a^b f v \, dx, \quad \forall v \in H_0^1(a,b).$$

Galerkin method

Principle 1: Make the residual orthogonal to a test space V.

Principle 2: Integrate by parts to lower smoothness requirements of the solution. (Seek a weak solution of the BVP).

Principle 3: Satisfy boundary conditions

- For Neumann boundary conditions, no restriction on test space, natural boundary conditions.
- For Dirichlet boundary conditions, remove boundary terms by restricting test space to have 0 on the boundaries. These are called essential boundary conditions.
- So we have the test space satisfying zero boundary conditions and the trial space satisfying specific boundary conditions for Dirichlet boundary conditions but no restrictions for Neumann boundary conditions.

Those 3 principles are already contained in the variational formulation.

Galerkin method (continue)

Principle 4: Choose trial and test space to be finite dimensional function spaces $U_N, V_N \subset H^1(a,b)$.

- For Dirichlet boundary conditions: $u_N = \phi_0 + \bar{u}$ where $\bar{u} \in V_N^0$ and ϕ_0 satisfies the boundary conditions. The space u_N exists in is an *affine space*. For V_N^0 choose a vector space of functions with zero boundary conditions, i.e. $V_N^0 \subset H_0^1(a,b)$.
- For Neumann boundary conditions $v \in V_N \subset H^1(a,b)$, $u \in U_N \subset H^1(a,b)$ where U_N and V_N are finite dimensional vector spaces of functions.

Galerkin equation

Principles 1 to 4 combine to give a Galerkin method.

So if we look back at our original problem, let $u = \sum_{j=1}^{N} c_j \phi_j(x)$.

$$
\begin{aligned}
&(Du', v') + (qu, v) = (f, v) + D(b)u'(b)v(b) - D(a)u'(a)v(a)\, \forall v \in V_N \\
\implies &(Du', \phi_k') + (qu, \phi_k) = (f, \phi_k) + D(b)u'(b)\phi_k(b) - D(a)u'(a)\phi_k(a) \\
\implies &(D\sum_{j=1}^{N} c_j \phi_j'(x), \phi_k') + (q\sum_{j=1}^{N} c_j \phi_j(x), \phi_k) = (f, \phi_k) + B_k \quad k = 1, \ldots, N.
\end{aligned}
$$

So we have a system of equations $A\vec{c} = \vec{f}$, where

$$
\begin{aligned}
A_{kj} &= \int_a^b D(x)\phi_k'(x)\phi_j'(x) + q(x)\phi_k(x)\phi_j(x)\, dx, \\
\vec{c} &= (c_1, \ldots, c_N)^T, \\
f_k &= \int_a^b f(x)\phi_k(x)\, dx + B_k.
\end{aligned}
$$

These are the *Galerkin equations*. We call $\vec{f}$ the (global) load vector.

Subspace conditions

Recall that first derivatives of H^1 function need not be continuous.

Theorem

A piecewise infinitely differentiable function $v : (a, b) \to \mathbb{R}$ belongs to $H^k(a, b)$ if and only if $v \in C^{k-1}(a, b)$.

Note that the theorem is also true in higher dimensions.

Choice of finite element dimensional space V_N:

- Spaces with global support e.g. polynomials or trigonometric functions. These are called Spectral Galerkin methods.
- Piecewise polynomial spaces. So ϕ_j will have local support. We call these spaces finite element spaces.

Finite element space

In FEM we construct V_h using piecewise polynomials in C_0. Denote the finite element space by X_h^k.

For X_h^k, each piece is a polynomial of degree k

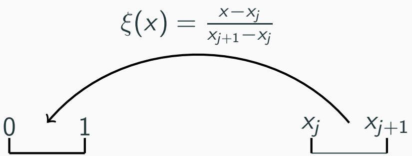
1. $n(k+1)$ parameters

n(k+1)	parameters
- (n-1)	match conditions at mesh points to ensure C^0 For
nk+1	parameters = dimension of X_h^k

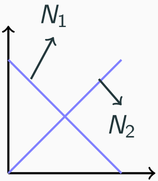
$X_{h,0}^k = X_h^k \cap H_0^1(a,b)$, zero value at 2 boundaries $\Rightarrow nk-1$ parameters.

See $nk+1$ basis functions. Degree of freedom (DOF $= nk+1$).

Construction of linear basis function



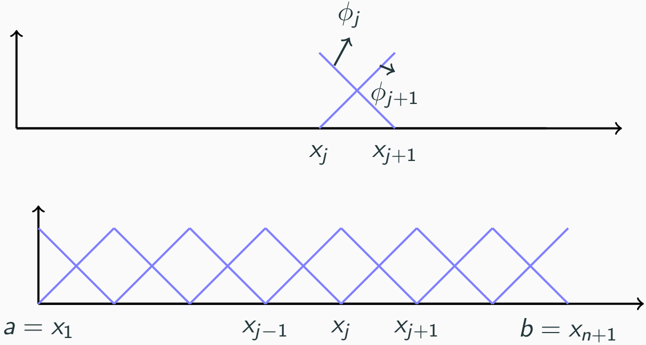
The reference element has 2 basis functions called shape function.



Linear functions with cardinal property $N_1(0) = 1, N_1(1) = 0$ and $N_2(0) = 0, N_2(1) = 1$
$\Rightarrow N_1(\xi) = 1 - \xi, N_2(\xi) = \xi$.

Global linear basis functions

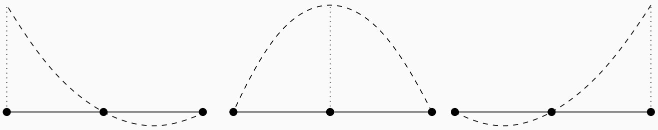
Under the map and the shape function, we get 2 basis functions defined on each element $E_j = (x_j, x_{j+1})$: $\phi_j = N_1(\xi(x))$ and $\phi_{j+1} = N_2(\xi(x))$



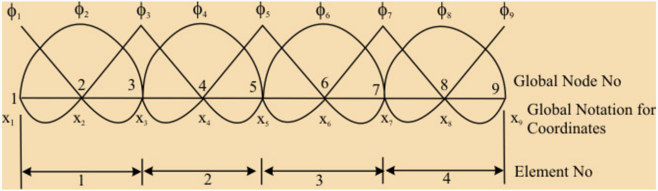
$\Rightarrow$ $(n+1)$ basis fun $\{\phi_j\} \in X_h^1$ (Linear Lagrange element) $\Rightarrow \phi_j(x_i) = \delta_{ij}$ and local support, i.e. $\phi_j \neq 0$ only at $K_{j-1} \cup K_j$.

Construction of quadratic basis function

$k = 2$, $2n + 1$ parameters $\Rightarrow 2n + 1$ DOFs $\Rightarrow$ (n+1) mesh points + n extra nodes



Shape functions on master element: $N_1 = (1 - \xi)(1 - 2\xi)$, $N_2 = 4(1 - \xi)\xi$ and $N_3 = \xi(2\xi - 1)$.



Implementation of finite element method

FEM implementations differ from others in 2 ways:

1. The stiffness/mass matrices and load vector are computed by doing integrals over each element, called *element stiffness matrices/load vectors*, and assembled into the final global matrices/vectors.
2. Rather than treat Dirichlet boundary conditions using general functions with $u = \phi_0 + \bar{u} \quad (\bar{u} \in V_h^0)$. Choose $\phi_0 \in X^k \backslash X_0^k$ so $u \in X^k$ with non zero end basis functions. We do integrals using $u = \sum_{j=1}^{n+1} u_h \phi_j$ then move the corresponding known Dirichlet node values to the right hand side of $A\vec{u} = \vec{b}$.

Implementation of linear finite element method

For linear Lagrange element, we have n elements and $n+1$ nodes giving global matrices of size $(n+1) \times (n+1)$. For each element we provide a list of nodes that make up the element in 1D so K_ℓ has nodes X_ℓ and $X_{\ell+1}$. The stiffness matrix and load vector are

$$K_{ij} = \int_a^b D(x)\phi_i'\phi_j'\,dx, \quad \text{for} \quad i,j = 1,\ldots,\text{degrees of freedom}$$

$$F_i = \int_a^b f(x)\phi_i(x)\,dx + \text{boundary terms},$$

Observation

$K_{ij} = 0$ unless ϕ_i' and ϕ_j' overlap so there are a lot of 0 terms. Instead we break things over the elements E_k:

$$\begin{aligned} K_{ij} &= \int_a^b D(x)\phi_i'\phi_j'\,dx \\ &= \sum_{\ell=1}^{n} \int_{x_\ell}^{x_{\ell+1}} D(x)\phi_i'\phi_j'\,dx \\ &= \sum_{\ell=1}^{n} \int_{E_\ell} D(x)\phi_i'\phi_j'\,dx, \end{aligned}$$

where each individual term in the sum is an element stiffness matrix. Similarly

$$F_i = \sum_{\ell=1}^{n} \int_{E_\ell} f(x)\phi_i\,dx.$$

Observation (continue)

The only basis functions that are non zero on element ℓ are ϕ_ℓ and $\phi_{\ell+1}$.

$$F_{E_\ell} = \begin{bmatrix} \int_{E_\ell} f(x)\phi_\ell \, dx \\ \int_{E_\ell} f(x)\phi_{\ell+1} \, dx \end{bmatrix},$$

$$K_{E_\ell} = \begin{bmatrix} \int_{E_\ell} D(x)\phi_\ell'^2 \, dx & \int_{E_\ell} D(x)\phi_\ell'\phi_{\ell+1}' \, dx \\ \int_{E_\ell} D(x)\phi_{\ell+1}'\phi_\ell' \, dx & \int_{E_\ell} D(x)\phi_{\ell+1}'^2 \, dx \end{bmatrix}.$$

Assemble load vector

$$\vec{f} = \begin{bmatrix} (f,\phi_1) \\ (f,\phi_2) \\ \vdots \\ \vdots \\ (f,\phi_{n+1}) \end{bmatrix} = \begin{bmatrix} \int_{x_1}^{x_2} f\phi_1 \\ \int_{x_1}^{x_3} f\phi_2 \\ \vdots \\ \vdots \\ \int_{x_n}^{x_{n+1}} f\phi_{n+1} \end{bmatrix} = \begin{bmatrix} \int_{x_1}^{x_2} f\phi_1 \\ \int_{x_1}^{x_2} f\phi_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \int_{x_2}^{x_3} f\phi_2 \\ \int_{x_2}^{x_3} f\phi_3 \\ 0 \\ \vdots \end{bmatrix} + \cdots + \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \int_{x_n}^{x_{n+1}} f\phi_n \\ \int_{x_n}^{x_{n+1}} f\phi_{n+1} \end{bmatrix}$$

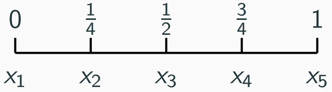
We only need compute element load vector F_{E_ℓ} and insert it into $(\ell : \ell+1, 1)$ locations in F.

Assemble stiffness matrix

$$K = \begin{bmatrix} K_{E_1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & K_{E_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & K_{E_n} \end{bmatrix}$$

We only need compute element load vector K_{E_ℓ} and insert it into $(\ell : \ell+1, \ell : \ell+1)$ locations in K.

Data structure for implementing linear FEM



$$\text{node} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{4} \\ \frac{1}{2} \\ \frac{3}{4} \\ 1 \end{bmatrix} \qquad \text{elem} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix}$$

elem is just the local numbering of local points to the global numbering of global points

Pseudo code

A = sparse(n+1, n+1);

b = zeros(n+1,1);

for j = 1:n

$x_j = node(elem(j, 1));$

$x_{j+1} = node(elem(j, 2));$

$K_{E_j} = elem\_stiff(x_j, x_{j+1}, D);$ % compute element stiffness matrix

$F_{E_j} = elem\_load(x_j, x_{j+1}, f);$ % compute element load vector

$A(elem(j, :), elem(j, :)) = A(elem(j, :), elem(j, :)) + K_{E_j};$

$b(elem(j, :), 1) = b(elem(j, :), 1) + F_{E_j};$

end

You should write funs to compute element matrices/vectors which can be done by mapping into master element.

Computation of elementary matrices/vector

Map each integral to the master element and then integrate over $[0,1]$.

$$\int_{x_j}^{x_{j+1}} f(x)\phi_j \, dx = h_j \int_0^1 f(x(\xi)) N_1(\xi)\, d\xi,$$
$$\int_{x_j}^{x_{j+1}} f(x)\phi_{j+1} \, dx = h_j \int_0^1 f(x(\xi)) N_2(\xi)\, d\xi.$$

Similarly,

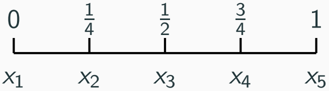
$$\begin{aligned}
&\int_{x_j}^{x_{j+1}} \frac{d\phi_j(x)}{dx}\frac{d\phi_{j+1}(x)}{dx}\, dx = \int_0^1 \frac{d\phi_j(x(\xi))}{dx}\frac{d\phi_{j+1}(x(\xi))}{dx}\frac{dx}{d\xi}\, d\xi \\
=&\int_0^1 \frac{dN_1(\xi)}{d\xi}\frac{d\xi}{dx}\frac{dN_2(\xi)}{d\xi}\frac{d\xi}{dx}\frac{dx}{d\xi}\, d\xi = \int_0^1 \frac{dN_1(\xi)}{d\xi}\frac{1}{h_j}\frac{dN_2(\xi)}{d\xi}\frac{1}{h_j} h_j \, d\xi \\
=&\frac{1}{h_j}\int_0^1 \frac{dN_1(\xi)}{d\xi}\frac{dN_2(\xi)}{d\xi}\, d\xi.
\end{aligned}$$

Elementary matrices/vector of quadratic element

$$F_{E_1} = \begin{bmatrix} (f, \phi_1) \\ (f, \phi_2) \\ (f, \phi_3) \end{bmatrix}, \quad F_{E_2} = \begin{bmatrix} (f, \phi_3) \\ (f, \phi_4) \\ (f, \phi_5) \end{bmatrix},$$

$$K_{E_1} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}, \quad K_{E_2} = \begin{bmatrix} K_{33} & K_{34} & K_{35} \\ K_{43} & K_{44} & K_{45} \\ K_{53} & K_{54} & K_{55} \end{bmatrix},$$

Data structure for implementing quadratic FEM



$$\text{node} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{4} \\ \frac{1}{2} \\ \frac{3}{4} \\ 1 \end{bmatrix} \qquad \text{elem} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} \qquad \text{elem2dof} = \begin{bmatrix} 1 & 6 & 2 \\ 2 & 7 & 3 \\ 3 & 8 & 4 \\ 4 & 9 & 5 \end{bmatrix}$$

elem2dof is just the numbering of local basis functions to the global numbering of global basis functions

Pseudo code of quadratic element

A = sparse(n+1, n+1);

b = zeros(n+1,1);

for j = 1:n

$x_j = node(elem(j, 1));$

$x_{j+1} = node(elem(j, 2));$

$K_{E_j} = elem\_stiffp2(x_j, x_{j+1}, D)$; %element stiffness matrix

$F_{E_j} = elem\_loadp2(x_j, x_{j+1}, f)$; % compute element load vector

$A(elem2dof(j, :), elem2dof(j, :)) = A(elem2dof(j, :), elem2dof(j, :)) + K_{E_j};$

$b(elem2dof(j, :), 1) = A(elem2dof(j, :), 1) + F_{E_j};$

end

You should write funs to compute element matrices/vectors which can be done by mapping into master element.

Galerkin orthogonality

The bilinear form (or inner product) defined by the ODE

$$a(u, v) = \int_a^b D(x)u'v' + q(x)uv \; dx,$$

which also induced an norm called energy norm

$$\|v\|_E^2 = \int_a^b D(x)|v'|^2 + q(x)v^2 \, dx.$$

Galerkin orthogonality:

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in V_h,$$

Properties of bilinear form

Two properties:

- Coercivity: $c\|u\|^2_{H^1(a,b)} \le a(u,u)$ (Poincaré's Lemma).
- Continuity: $a(u,v) \le C\|u\|_{H^1(a,b)}\|v\|_{H^1(a,b)}$ (Cauchy-Schwartz Inequality)

Lemma : (Lax-Miligram)

Let $a(\cdot,\cdot)$ be a bounded coercive and continuous bilinear form on a Hilbert space $H_0^1(a,b)$. Then for any function $f \in L^2(a,b)$, there exists a unique u in $H_0^1(a,b)$ such that

$$a(u,v) = (f,v), \quad v \in H_0^1(a,b).$$

Sufficient condition for this weak formulation to have a unique solution.

Ceà's Lemma

Lemma

Let u_h be the finite element solution of the model problem. Assume V_h is a subspace of H^1, then we have

$$\|u - u_h\|_E = \min_{w_h \in V_h} \|u - w_h\|_E,$$

Best approximation property: the Galerkin solution is the best approximation to u from all the functions in V_h.

Error estimate

Theorem

Assume $u \in H^s$ for $s \geq 2$ and $u_h \in V_h = X_h^k$. Then:

$$\|u - u_h\|_{H_0^1} \leq Ch^l \|u\|_{H^{l+1}},$$

where $l = \min\{k, s-1\}$.

Also,

$$\|u - u_h\|_{L^2} \leq Ch^{l+1} \|u\|_{H^{l+1}}.$$

Assume u is smooth enough. kth order in H^1 norm and $(k+1)$th order in L^2 norm.

End of week 3!