MAST90026 Computational Differential Equations: Week 6

Jesse Collis

Modified from Hailong Guo (2022)

Semester 1 2024

The University of Melbourne

How do we solve a PDE using Finite Element Methods?

1. Transform the PDE to its weak form.
2. Choose test/trial spaces (these depend on the boundary conditions).

3. Form the Galerkin equations.
4. Choose a mesh.
5. Choose finite element space on the mesh (which will generate a sparse linear system which is solved to get the solution).

Note: Combine step 1 and step 2 into weak formulation.

Weak formulation

The weak formulations are defined by: a space $\boldsymbol{X}$; a bilinear form a; a linear form ℓ.

The weak formulations identify

- ESSENTIAL boundary conditions, Dirichlet - reflected in $\boldsymbol{X}$;
- NATURAL boundary conditions, Neumann/Robin - reflected in a, ℓ.

The Dirichlet problem

Consider the equation:

$$-\nabla \cdot (D(x,y)\nabla u(x,y)) + q(x,y)u(x,y) = f(x,y),$$

where $D > 0$, $q \geq 0$.

(If $q = 0$ then we have the Poisson equation, and if both $q = 0$ and $f = 0$ then we have the Laplace equation.)

The Dirichlet boundary condition is

$$u = 0, \quad (x,y) \in \partial\Omega.$$

The Dirichlet problem: weak formulation

Let

$$a(w,v) = \int_\Omega D(x,y)\nabla w \nabla v dA + \int_\Omega q w v dA, \quad \forall w, v \in X,$$

a bilinear form and

$$\ell(v) = \int_\Omega f v dA, \quad \forall v \in X,$$

a linear form.

Weak form: find $u \in X$ such that

$$a(u,v) = \ell(v), \quad \forall v \in X.$$

The Dirichlet problem: solution space

Since a involves only first derivatives,

$$\boldsymbol{X} = \left\{ \boldsymbol{v} \in \boldsymbol{H}^1(\Omega) | v|_\Gamma = 0 \right\} \equiv \boldsymbol{H}_0^1(\Omega):$$

$$\boldsymbol{H}^1(\Omega) \equiv \left\{ v \mid \int_\Omega v^2 d\boldsymbol{A}, \int_\Omega v_x^2 d\boldsymbol{A}, \int_\Omega v_y^2 d\boldsymbol{A} \text{ finite } \right\}$$

$$\underbrace{(w, v)_{H^1(\Omega)}}_{\text{inner product}} = \int_\Omega \nabla w \cdot \nabla v + wvdA;$$

$$\underbrace{\|w\|_{H^1(\Omega)}}_{\text{norm}} = \left(\int_\Omega |\nabla w|^2 + w^2 dA \right)^{1/2}$$

The Dirichlet problem: well-posedness

Suppose $D(x) \geq c > 0$ and $q(x) \geq 0$:

- Coercivity: $c\|u\|^2_{H^1_0(a,b)} \leq a(u,u)$ (Poincaré's Lemma).
- Continuity: $a(u,v) \leq C\|u\|_{H^1_0(a,b)}\|v\|_{H^1_0(a,b)}$ (Cauchy-Schwartz Inequality)

The Lax-Miligram Lemma implies weak formulation to have a unique solution.

The Neumann problem

Consider the equation:

$$
\begin{aligned}
-\nabla \cdot (D(x,y)\nabla u(x,y)) + q(x,y)u(x,y) = f(x,y), &\quad (x,y) \in \Omega, \\
D\frac{\partial u}{\partial n}(x,y) = g(x,y), &\quad (x,y) \in \partial\Omega.
\end{aligned}
$$

where $D > 0$, $q > 0$.

The Neumann problem: weak formulation

Let

$$a(w,v) = \int_\Omega (D(x,y)\nabla w \dot{\nabla} v dA + \int_\Omega q w v dA, \quad \forall w, v \in X,$$

a bilinear form and

$$\ell(v) = \int_\Omega f v dA + \int_{\partial\Omega} g v dS, \quad \forall v \in X,$$

a linear form.

Weak form: find $u \in X = H^1(\Omega)$ such that

$$a(u,v) = \ell(v), \quad \forall v \in X.$$

The Nuemann problem: well-posedness

Suppose $D(x) \geq c > 0$ and $q(x) > 0$:

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

The Lax-Miligram Lemma implies weak formulation to have a unique solution.

Inhomogeneous Dirichlet condition

let $u = G + \bar{u}$, where $\bar{u} \in H_0^1$ and G is a function defined on Ω which satisfies the boundary, conditions $u = g$ on $\partial\Omega$.

The weak formulation becomes

$$\int (D\nabla \bar{u} \cdot \nabla v + q\bar{u}v)\, dA = \int fv\, dA + \underbrace{\int_{\partial\Omega} D\frac{\partial u}{\partial n} v\, ds}_{= 0 (v = 0 \text{ on } \partial\Omega)} - \int_{\Omega} (D\nabla G \cdot \nabla v + qGv)\, dA$$

Remark: only need the above form for theoretical analysis purpose.

Essential vs Natural

Essential boundary conditions: Imposed by $\boldsymbol{X}$.

Natural boundary conditions: Imposed by a, ℓ.

Here:

> Essential $\Leftrightarrow$ Dirichlet ($v|_{\partial\Omega} = 0$).
>
> Natural $\Leftrightarrow$ Neumann/Robin ($v|_{\partial\Omega}$ unrestricted) .

Important theoretical and numerical ramifications.

Galerkin equation

Select nodal basis $\{\phi_i\}$, which gives us the Galerkin equations

$$\sum_j u_j \int D\nabla\phi_j \cdot \nabla\phi_i + q\phi_i\phi_j \, dA = \int f\phi_i \, dA + \text{boundary terms}, \qquad \forall i.$$

Obtain the system of linear equations $Au = F$, where, like the 1D case, we have $A = K + M$.

Mesh generation

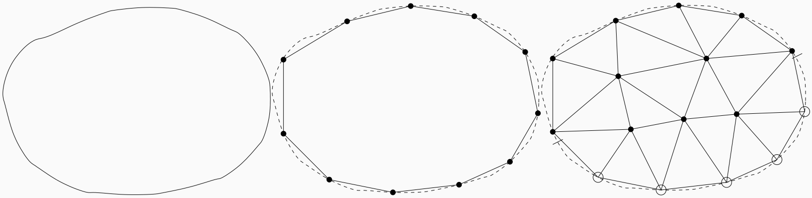
In order to specify the nodal basis, we need to discretise the domain Ω, to a mesh Ω_h. There are many options:

- Triangulation;
- Rectangles (rectangular domain); or
- Others like:
  - Triangles plus rectangles; or
  - Quadrilaterals;
- (In 3D: Tetrahedra or bricks.)

Treatment of curve domain

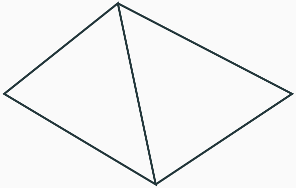
To treat curved boundaries we can either:

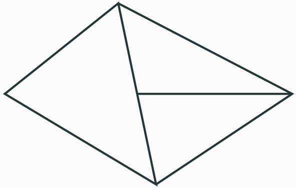
- Approximate the domain by a polygon. This gives $O(h)$ error in the energy norm, which is acceptable for P_1 (i.e. linear elements) but not for P_2 (i.e. quadratic)).
- Use a different kind of element (isoparametric), which has $O(h^2)$ error in the energy norm.



For any polygonal domain Ω, $\Omega = \cup_j T_j$. The set of all elements are denoted by $\mathcal{T}_h$.

Conforming and non conforming triangles

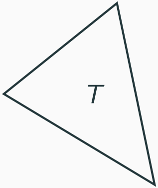




Conforming triangulation $\mathcal{T}_h$: $T_i \cap T_j \neq \emptyset \Rightarrow$ the intersection is either a vertex of T_i and T_j or the whole edge of T_i and T_j.

Linear finite element

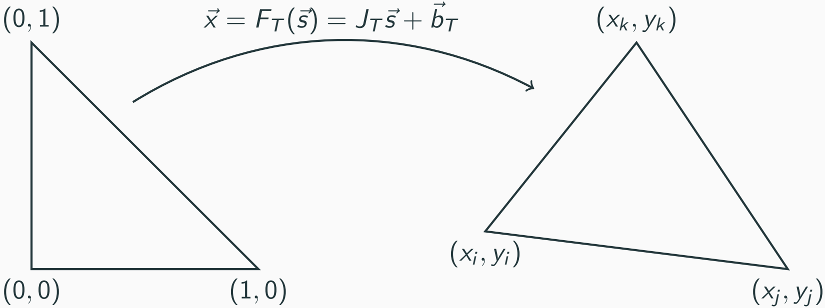
$$\boldsymbol{X}_h = \{ \underbrace{v \in \boldsymbol{X}}_{v \in C^0(\Omega)} \, | v|_{\boldsymbol{T}} \in \mathbb{P}_1(\boldsymbol{T}), \forall \boldsymbol{T} \in \mathcal{T}_h \}$$



$$\mathbb{P}_1(T) : \, v\big|_{T_h} = c_1 + \underbrace{c_2}_{v_x} x + \underbrace{c_3}_{v_y} y, \quad c_1, c_2, c_3 \in \mathbb{R}$$

Need 3 basis functions per element. Need 3 DOF per element.

Mapping from canonical triangle to arbitrary element



Construction of affine mapping

Mapping condition:

- $F_T((0,0)) = (x_i, y_i)$,
- $F_T((1,0)) = (x_j, y_j)$,
- $F_T((0,1)) = (x_k, y_k)$.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_j - x_i & x_k - x_i \\ y_j - y_i & y_k - y_i \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\Rightarrow \vec{x} = F_T(\vec{s}) = J_T \vec{s} + \vec{b}_T.$$

Nodal basis

Nodal basis at reference element $N_j(v_i) = \delta_{ij}$(where v_i are three vertices of reference element):

$$
\begin{aligned}
N_1(\xi,\eta) =& 1-\xi-\eta, & \nabla_{\vec{s}} N_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \\
N_2(\xi,\eta) =& \xi, & \nabla_{\vec{s}} N_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
N_3(\xi,\eta) =& \eta, & \nabla_{\vec{s}} N_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.
\end{aligned}
$$

Nodal basis at general element: $\phi_j(x,y) = N_j(F_T^{-1}(x,y))$.

Evaluation of basis function at general element

Observation: $\phi_j(x, y) = N_j(F_T^{-1}(x, y)) \Leftrightarrow \phi_j(F_T(\xi, \eta)) = N_j(\xi, \eta)$.

Chain rule:

$$\frac{\partial N_j}{\partial \xi} = \frac{\partial}{\partial \xi}\phi_j(F_T(\xi, \eta)) = \frac{\partial \phi_j}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial \phi_j}{\partial y}\frac{\partial y}{\partial \xi} = \frac{\partial \phi_j}{\partial x}J_{11} + \frac{\partial \phi_j}{\partial y}J_{21}$$

Similarly,

$$\frac{\partial N_j}{\partial \eta} = \frac{\partial}{\partial \eta}\phi_j(F_T(\xi, \eta)) = \frac{\partial \phi_j}{\partial x}J_{12} + \frac{\partial \phi_j}{\partial y}J_{22}$$

Combining those, implies

$$\begin{bmatrix} \frac{\partial N_j}{\partial \xi} \\ \frac{\partial N_j}{\partial \eta} \end{bmatrix} = \begin{bmatrix} J_{11}\frac{\partial \phi_j}{\partial x} + J_{21}\frac{\partial \phi_j}{\partial y} \\ J_{12}\frac{\partial \phi_j}{\partial x} + J_{22}\frac{\partial \phi_j}{\partial y} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{21} \\ J_{12} & J_{22} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_j}{\partial x} \\ \frac{\partial \phi_j}{\partial y} \end{bmatrix}$$

In other words, $\nabla_{\vec{s}} N_j = J_T^T \nabla \phi_j \Rightarrow \nabla \phi_j = J_T^{-T} \nabla_{\vec{s}} N_j$.

Computation of element load vector

Element load vector:

$$f_T = \begin{bmatrix} \int_T f\phi_i \, dA \\ \int_T f\phi_j \, dA \\ \int_T f\phi_k \, dA \end{bmatrix},$$

Calculate the entries as

$$\begin{aligned} \int_T f\phi_i \, dA &= \int_T f(\vec{x})\phi_j(\vec{x}) \, dxdy \\ &= \int_{T_R} f(\vec{x}(\vec{s}))N_j(\vec{s})|\det(J_T)| \, d\xi d\eta. \end{aligned}$$

Computation of element stiffness matrix

Element stiffness matrix,

$$K_T = \begin{bmatrix} \int_T D\nabla\phi_i \cdot \nabla\phi_i \, dA & \int_T D\nabla\phi_i \cdot \nabla\phi_j \, dA & \int_T D\nabla\phi_i \cdot \nabla\phi_k \, dA \\ \int_T D\nabla\phi_j \cdot \nabla\phi_i \, dA & \int_T D\nabla\phi_j \cdot \nabla\phi_j \, dA & \int_T D\nabla\phi_j \cdot \nabla\phi_k \, dA \\ \int_T D\nabla\phi_k \cdot \nabla\phi_i \, dA & \int_T D\nabla\phi_k \cdot \nabla\phi_j \, dA & \int_T D\nabla\phi_k \cdot \nabla\phi_k \, dA \end{bmatrix},$$

Calculate the entries as

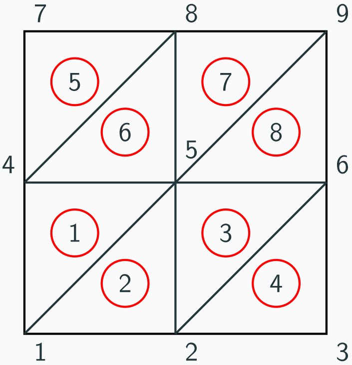
$$\begin{aligned} \int_T D\nabla\phi_i \cdot \nabla\phi_j \, dA &= \int_T D(\vec{x})\nabla\phi_i(\vec{x}) \cdot \nabla\phi_j(\vec{x}) \, dxdy \\ &= \int_{T_R} D(\vec{x}(\vec{s})) \left(J_T^{-T}\nabla_{\vec{s}} N_i(\vec{s})\right) \cdot \left(J_T^{-T}\nabla_{\vec{s}} N_j(\vec{s})\right) |\det(J_T)| \, d\xi d\eta. \end{aligned}$$

Quadrature rules on triangle

Some quadrature rules on reference triangles T_R:

- $\int_{T_R} g(\xi, \eta) d\xi d\eta \approx \frac{1}{2} g(1/3, 1/3)$,
- $\int_{T_R} g(\xi, \eta) d\xi d\eta \approx \frac{1}{6} \left[g(1/2, 0) + g(1/2, 1/2) + g(0, 1/2) \right]$,

Data structure for implementing linear 2D FEM



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

node and *elem* can be also generated using mesh generator like **distmesh** and **triangle**.

Pseudo code

N = size(node,1); NT = size(elem,1)

A = sparse(N, N); b = zeros(N,1);

for j = 1:NT

$P1 = node(elem(j, 1), :);$

$P2 = node(elem(j, 2), :);$

$P3 = node(elem(j, 3), :);$

$K_{E_j} = elem\_stiff(P1, P2, P3, D)$; % compute local stiffness matrix

$F_{E_j} = elem\_load(P1, P2, P3, f)$; % compute local 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 functions to compute element matrices/vectors which can be done by mapping into master element.

Impose Dirichlet boundary condition

Need identify boundary nodes.

On rectangle domain $[a, b] \times [c, d]$:

$isLeftBnd = abs(node(:, 1) - a) < eps$

$isRightBnd = abs(node(:, 1) - b) < eps$

$isButtomBnd = abs(node(:, 2) - c) < eps$

$isTopBnd = abs(node(:, 2) - d) < eps$

Impose Dirichlet boundary condition (continue)

$isBndNode = false(N, 1);$

$isBndNode(isLeftBnd) = true;$

$isBndNode(isRightBnd) = true;$

$isBndNode(isButtomBnd) = true;$

$isBndNode(isTopBnd) = true;$

$bndNode = find(isBndNode);$

$freeNode = find(\sim isBndNode);$

$u = zeros(N, 1);$

$u(bndNode) = g_D(node(bndNode, :));$

$b = b - A * u;$

$u(freeNode) = A(freeNode, freeNode) \backslash b(freeNode);$

Other useful **MATLAB** command: *trimesh*, *triplot*, and *trisurf*.

Theoretical analysis: Energy norm

From the bilinear form we define an energy norm

$\|v\|_E^2 \equiv a(v, v)$. Then

$$\|u - u_h\|_E = \inf_{w_h \in X_h} \|u - w_h\|_E$$

u is the solution to the PDE and u_h is the projection of u on $\boldsymbol{X}_h$ with respect to the bilinear form.

inf is the infimum, i.e., the maximal lower bound.

Theoretical analysis: $H1$ norm

Recall $\|v\|^2_{H^1(\Omega)} = \int_\Omega |\nabla v|^2 + v^2 dA$ Then

$$\|u - u_h\|_{H^1(\Omega)} \leq \left(1 + \frac{\beta}{\alpha}\right) \inf_{w_h \in X_h} \|u - w_h\|_{H^1(\Omega)}$$

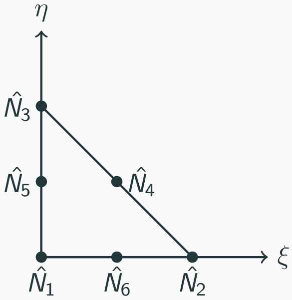
α: coercivity constant (> 0); β: continuity constant ($= 1$).

Theoretical analysis: $H1$ and $L2$ norm

For $f \in L^2(\Omega)$ and Ω convex,

$$\begin{aligned} \|u - u_h\|_E &\leq Ch\|u\|_{H^2(\Omega)} \\ \|u - u_h\|_{H^1(\Omega)} &\leq Ch\|u\|_{H^2(\Omega)} \\ \text{and} \quad \|u - u_h\|_{L^2(\Omega)} &\leq Ch^2\|u\|_{H^2(\Omega)} \end{aligned}$$

Quadratic Lagrange element



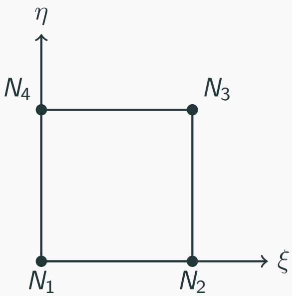
N_j shape function for linear element.

$$\hat{N}_1 = N_1(1 - 2N_1) \quad \hat{N}_4 = 4N_2N_3$$

$$\hat{N}_2 = N_2(1 - 2N_2) \quad \hat{N}_5 = 4N_1N_3$$

$$\hat{N}_3 = N_3(1 - 2N_3) \quad \hat{N}_6 = 4N_1N_2$$

Bilinear element Q1



$$N_1 = (1-\xi)(1-\eta)$$

$$N_2 = (1-\xi)\eta$$

$$N_3 = \xi(1-\eta)$$

$$N_3 = \xi\eta$$

End of week 6!