FEM12 - lecture 2

Johan Jansson

jjan@csc.kth.se

CSC

KTH

Weak formulation

Variational/Weak formulation: multiply by test function and integrate:

$$\int_0^1 R(u)v dx = \int_0^1 (-(au')' - f)v dx = 0, \quad \forall v \in V$$

$$V = \left\{ v : \int_0^1 v^2 \, dx < C, \; \int_0^1 (v')^2 \, dx < C, \; v(0) = v(1) = 0 \right\},$$

Exercise (section 8.1.2): explain why $\int_0^1 R(u)vdx = 0 \Rightarrow R(u) = 0$ for continuous a and u.

Galerkin's method

$$(R(U), v)_{L_2} = 0, \quad x \in [a, b], \quad \forall v \in V_h$$

But we have:

$$(R(u), v) = \int_0^1 (-(au')' - f) v dx = 0$$

U is not compatible (only has one derivative).

Technical step:

Integrate by parts (move derivative to test function)

- Piecewise linear approximation only has one derivative
- Simplifies enforcement of boundary conditions

Galerkin's method

Recall integration by parts (fundamental theorem):

$$\int_0^1 w'v dx = -\int_0^1 wv' dx + w(1)v(1) - w(0)v(0)$$

$$R(u) = -(au')' - f$$

$$(R(u), v) = \int_0^1 -(au')'v - fv dx = [w = au'] =$$

$$\int_0^1 (au')v' - fv dx + au'(1)v(1) - au'(0)v(0)$$

Boundary conditions:
For homogenous Dirichlet BC we can use $v(a) = v(b) = 0$
For homogenous Neumann BC we have $-au' = 0$

Galerkin's method

Insert piecewise linear approximation:

$$U(x) = \sum_{j=1}^{M} \xi_j \phi_j(x)$$

We are left to solve:

$$\int_a^b (aU')v' - fv dx = 0, \quad x \in [a,b], \quad \forall v \in V_h$$

Or equivalently:

$$\int_a^b (aU')\phi_i' - f\phi_i dx = 0,$$
$$x \in [a,b], \quad i = 1, ..., M$$

Discrete system

Substituting U:

$$\int_a^b (a(\sum_{j=1}^{M} \xi_j \phi_j)')\phi_i' - f\phi_i dx = 0,$$
$$x \in [a, b], i = 1, ..., M$$

Clean up:

$$\sum_{j=1}^{M} \int_a^b a\xi_j \phi_j' \phi_i' - f\phi_i dx = 0,$$
$$x \in [a, b], i = 1, ..., M$$

Discrete system

Left with algebraic system in $\xi = (\xi_1, ..., \xi_M)^\top$:

$$F(\xi) = 0$$

In this case F is a linear system $F(\xi) = A\xi - b = 0$ with:

$$A_{ij} = \sum_{j=1}^{M} \int_a^b a\phi_j' \phi_i' dx,$$

$$b_i = \int_a^b -f\phi_i dx$$

Solve $A\xi = b$, costruct solution function $U(x) = \sum_{j=1}^{M} \xi_j \phi_j(x)$

If F is not linear, can use Newton's method.

Discrete system (work in groups)

Exercise: 6.8, 6.9 and 6.10 (explain computation of matrix and vector entries)

Piecewise polynomials in 2D

Construct triangulation T of domain Ω

Define size of triangle $K \in T$ is h_K as diameter of triangle

Define N as node (in this case vertex of triangle)

Want to define basis functions for vector space V_h: space of piecewise linear functions on T

Requirement for nodal basis:

$$\phi_j(N_i) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \qquad i, j = 1, ..., M \tag{1}$$

Piecewise polynomials in 2D

Define local basis functions v^i on triangle K with vertices $a^i = (a_1^i, a_2^i)$, $i = 1, 2, 3$

v is linear $\Rightarrow v(x) = c_0 + c_1 x_1 + c_2 x_2$

Values of v in vertices: $v_i = v(a^i)$ (1 or 0)

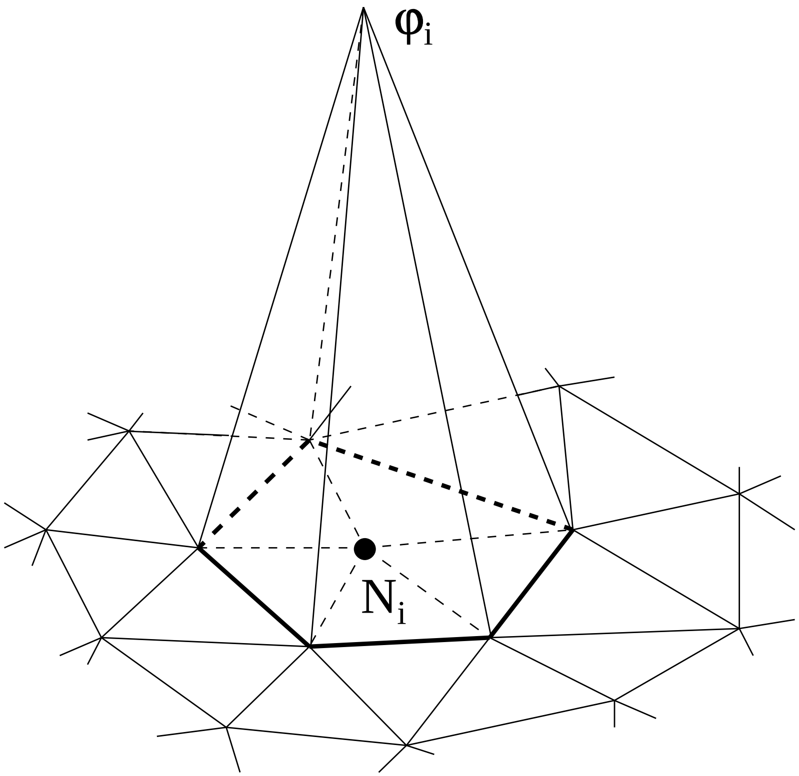
Linear system for coefficients c:

$$\begin{pmatrix} 1 & a_1^1 & a_2^1 \\ 1 & a_1^2 & a_2^2 \\ 1 & a_1^3 & a_1^3 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

Piecewise polynomials in 2D

Sum local basis functions:

$$\phi_i = \sum_j v^j, \quad N_i = a_j \tag{2}$$



Poisson in 2D (demo)

Polynomial interpolation

We assume that f is continuous on $[a, b]$ and choose distinct interpolation nodes $a \leq \xi_0 < \xi_1 < \cdots < \xi_q \leq b$ and define a polynomial interpolant $\pi_q f \in \mathcal{P}^q(a, b)$, that interpolates $f(x)$ at the nodes $\{\xi_i\}$ by requiring that $\pi_q f$ take the same values as f at the nodes, i.e. $\pi_q f(\xi_i) = f(\xi_i)$ for $i = 0, ..., q$. Using the Lagrange basis corresponding to the ξ_i, we can express $\pi_q f$ using “Lagrange’s formula”:

$$\pi_q f(x) = f(\xi_0)\lambda_0(x) + f(\xi_1)\lambda_1(x) + \cdots + f(\xi_q)\lambda_q(x) \quad \text{for } a \leq x \leq b$$

General bilinear form $a(\cdot, \cdot)$

In general the matrix A_h, representing a bilinear form

$$a(u, v) = (A(u), v),$$

is given by

$$(A_h)_{ij} = a(\varphi_j, \hat{\varphi}_i).$$

and the vector b_h representing a linear form

$$L(v) = (f, v),$$

is given by

$$(b_h)_i = L(\hat{\varphi}_i).$$

Assembling the matrices

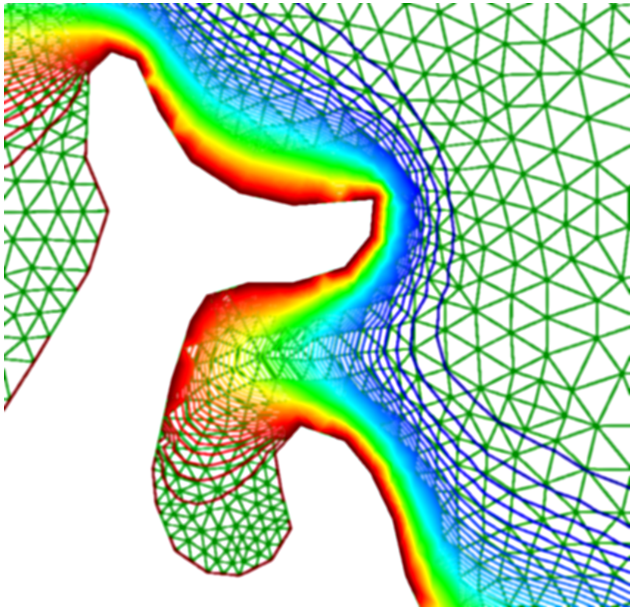
Computing $(A_h)_{ij}$

Note that

$$(A_h)_{ij} \quad = \quad a(\varphi_j, \hat{\varphi}_i) = \sum_{K \in \mathcal{T}} a(\varphi_j, \hat{\varphi}_i)_K .$$

Iterate over all elements K and for each element K compute the contributions to all $(A_h)_{ij}$, for which φ_j and $\hat{\varphi}_i$ are supported within K.

Assembly of discrete system



Noting that $a(v,u) = \sum_{K\in\mathcal{T}} a_K(v,u)$, the matrix A can be assembled by

$A = 0$
for all elements $K \in \mathcal{T}$
    A += A^K

The *element matrix* A^K is defined by

$$A^K_{ij} = a_K(\hat{\phi}_i, \phi_j)$$

for all local basis functions $\hat{\phi}_i$ and ϕ_j on K

Assembling A_h

for all elements $K \in \mathcal{T}$

for all test functions $\hat{\varphi}_i$ on K

for all trial functions φ_j on K

1. Compute $I = a(\varphi_j, \hat{\varphi}_i)_K$
2. Add I to $(A_h)_{ij}$

end

end

end

Assembling b

for all elements $K \in \mathcal{T}$

for all test functions $\hat{\varphi}_i$ on K

1. Compute $I = L(\hat{\varphi}_i)_K$
2. Add I to b_i

end

end

L_2 projection

We seek a polynomial approximate solution $U \in P^q(a,b)$ to the equation:

$$R(u) = u - f = 0, \quad x \in (a,b)$$

where f in general is not polynomial, i.e. $f \notin P^q(a,b)$.
This means $R(U)$ can in general not be zero. The best we can hope for is that $R(U)$ is orthogonal to $P^q(a,b)$ which means solving the equation:

$$(R(U), v)_{L_2} = (U - f, v)_{L_2} = 0, \quad x \in \Omega, \quad \forall v \in P^q(a,b)$$

Error estimate

The orthogonality condition means the computed L_2 projection U is the best possible approximation of f in $P^q(a,b)$ in the L_2 norm:

$$
\begin{aligned}
\|f-U\|^2 = (f-U, f-U) = \\
& (f-U, f-v) + (f-U, v-U) = \\
& [v-U \in P^q(a,b)] = (f-U, f-v) \leq \|f-U\|\|f-v\| \\
& \Rightarrow
\end{aligned}
$$

$$
\|f-U\| \leq \|f-v\|, \quad \forall v \in P^q(a,b)
$$

Error estimate

Since $\pi f \in P^q(a, b)$, we can choose $v = \pi f$ which gives:

$$\|f - U\| \leq \|f - \pi f\|$$

i.e. we can use an interpolation error estimate since it bounds the projection error.