Problem Session Notes for Finite Element Method

Hanfeng Zhai

Department of Mechanical Engineering
Stanford University

Disclaimer: These notes are intended solely for personal reference and study purposes. They represent my own understanding of the course material and may contain errors or inaccuracies. The content presented here should not be considered as an authoritative source, and reliance solely on these materials is not recommended. If you notice any materials that potentially infringe upon the copyright of others, please contact me at hzhai@stanford.edu so that appropriate action can be taken. Your feedback is greatly appreciated.

Problem Session #1 H. Zhai

1/2/2025. (hzhai@stanford.edu)

Overview

▷ Problem 1.1. recall strong form.

$$\underbrace{-\left(k(x)\,u'(x)\right)' + b(x)\,u'(x) + c(x)\,u(x)}_{\mathcal{L}(u,\,x)} = f(x)$$

$\forall x \in \Omega$ ← domain.

(1) $\mathcal{L}(u, x) - f = \phi$

(2) $\hat{\mathcal{L}}(\hat{u}, x) - f \neq \phi$ ← approximated

Construct residual: $R_\Omega = (2) - \cancel{(1)}^{\,0} \neq \phi$.

Variational formulation → Integrate the residual:

$$\int_\Omega R_\Omega \, v \, d\Omega = \phi$$

v ↳ test functions (weighting func.)

Weak form can be constructed as.

$$\int_{\Omega} R_{\Omega}\, v \, d\Omega + \int_{\Gamma} R_{\Gamma}\, d\Gamma = \emptyset$$

$\int_{\Omega} R_{\Omega}\, v \, d\Omega$ $\downarrow$ Residual over domain

$\int_{\Gamma} R_{\Gamma}\, d\Gamma$ $\downarrow$ Residual over boundaries.

R_{Ω}: linear combinations of basis functions

$$\sum_{m=1}^{M} a_m N_m$$

Galerkin method.

$v(x)$: "can be any function of x that is sufficiently well behaved for the integrals to exist." ← you may put on constraints on $v(x)$ based on your problems.

$\Downarrow$

You will explore this in your HW 1.

...m LaForce
...NERGY 281
...006.

▸ Boundary Conditions.

- Dirichlet B.C.s. $u(x=a) = g_0$

- Neumann B.C.s $u'(x=b) = d_L$

- Robin B.C.s $u'(x=c) + u(x=c) = \alpha$

Trial space. $\mathcal{S} = \{ w : \Omega \to \mathbb{R} \text{ smooth} \}$.

Test space: $\mathcal{V} = \{ w : \Omega \to \mathbb{R} \text{ smooth} \}$

trial functions → approximation of the solution.

... represents the sol'n to the problem,

examples. polynomials: $u(x) = a + bx + cx^2 + \cdots$

test functions → <u>test</u> how well trial function

satisfies the governing equations

↳ PDE.

... used to evaluate the error.

Example (1.10)

$f : [a,b] \to \mathbb{R}$, find $u : [a,b] \to \mathbb{R}$ s.t.

$$u''' = f, \qquad x \in (a,b)$$
$$u(a) = 1$$
$$u'(b) = 2$$
$$u''(a) = 3.$$

Solution (Exact)

$$\int_a^x f(y)\,dy = \int_a^x u'''(y)\,dy$$
$$= u''(x) - u''(a) = u''(x) - 3.$$

$$\int_b^x \int_a^z f(y)\,dy\,dz = \int_b^x u''(z) - 3\,dz.$$
$$= u'(x) - u'(b) - 3(x-b)$$
$$= u'(x) - 2 - 3(x-b)$$

$$\int_a^x \int_b^w \int_a^z f(y)\,dy\,dz\,dw = \int_a^x u'(w) - 2 - 3(w-b)\,dw$$
$$= u(x) - \cancelto{1}{u(a)} - 2(x-a) - \frac{3}{2}(x^2-a^2)$$
$$+ 3b(x-a)$$

Exact solution writes.

$$u(x) = 1 + (2-3b)(x-a) + \frac{3}{2}(x^2 - a^2)$$
$$+ \int_a^x \int_b^w \int_a^z f(y)\, dy\, dz\, dw.$$

Solving it w/ variational method.

(a) form the residual. $r = u''' - f$.

(b) multiply by test function and integrate.

$$\int_a^b (u''' - f)\, v\, dx = 0$$

$\nwarrow$ Smooth

(c). Integration by parts.

$$u''(b)v(b) - u''(a)v(a) - \int_a^b u''v' + fv\, dx = 0$$

for all v <u>smooth</u>.

(d). Subs. B.C.s. we know $u''(a) = 3$.

... needs to request $v(b) = 0$

$$\rightarrow \; -3v(a) - \int_a^b u''v' + \int fv \, dx = 0.$$

Hence, $u''(a) = 3$ is a natural B.C.s

(e). formulate the weak form.

essential B.C.s: $u(a) = 1$ & $u'(b) = 2$.

let: $\mathcal{S} = \{u : [a,b] \rightarrow \mathbb{R} \text{ smooth} \mid u(a) = 1,\ u'(b) = 2\}$.

$\mathcal{V}^0 = \{v : [a,b] \rightarrow \mathbb{R} \text{ smooth} \mid v(b) = 0\}$

* Weak form of the problem.

find $u \in \mathcal{S}$ s.t. for all $v \in \mathcal{V}^0$.

$$-\int_a^b u''v' \, dx = \int_a^b fv \, dx - 3v(a)$$

Problem Session 2 1/16/2025

- ▷ Vector space.
- ▷ Euler - Lagrange
- ▷ (possibly) test functions, shape functions & basis functions ...

Vector Space

~ Renzo Cavalieri, CSO

"Nerdy definition" 

"A vector space is a set that is closed under addition and scalar multiplication".

basis for a vector space

↳ sets w/ simple structure

⇝ they can be **added** together & **multiplied** by scalars.

Definition.

Additive Multiplicative

Additive Closure $u+v \in V$

Additive Commutativity $u+v = v+u$

Additive Associativity $(u+v)+w = u+(v+w)$.

Zero. $u + 0_V = u \quad \forall u \in V$

Additive Inverse For every u, exists w $u+w = 0_V$

Multiplicative Closure $c \cdot v \in V$

Distributivity $(c+d) \cdot v = c \cdot v + d \cdot v$.

Distributivity $c \cdot (u+v) = c \cdot u + c \cdot v$

Associativity $(cd) \cdot v = c \cdot (d \cdot v)$

Unity $1 \cdot v = v \quad \forall v \in V$

Examples Credit : Sebastian Tomaskovic-Moore, UPenn.

① $\{(a,b) \in \mathbb{R}^2 : b = 3a+1\}$.

counter-example :
- No zero vector
- Not closed addition & multiplication.

② $\{(a,b) \in \mathbb{R}^2\}$ w/ scalar mul. $k(a,b) = (ka, b)$

$(r+s)(a,b) = ((r+s)a, b) = (ra+sa, b)$

$r(a,b) + s(a,b) = (ra, b) + (sa, b) = (ra+sa, 2b)$.

violates the distributivity !!!!!

③ $\{(a,b) \in \mathbb{R}^2\}$ w/ Scalar mul. $k(a,b) = (ka, 0)$.

$$1(a,b) = (1a, 0) = (a, 0) \neq (a,b)$$

violates both Mul. closure & Unity mul.

Euler Lagrange Equation

credit: Norbert Stoop, MIT

Let us define an "Energy functional".

$$P(u) = \int_0^1 F(u, u')\, dx \quad \text{w/} \quad \begin{cases} u(0) = a \\ u(1) = b \end{cases}$$

Recall functional derivative:

$$J[f] = \int_a^b L(x, f(x), f'(x))\, dx.$$

$$\frac{\delta J}{\delta f} = \frac{\partial L}{\partial f} - \frac{d}{dx}\frac{\partial L}{\partial f'} \longleftrightarrow \delta J = \int_a^b \left(\frac{\partial L}{\partial f}\, \delta f(x) + \frac{\partial L}{\partial f'} \frac{d}{dx} \delta f(x) \right) dx$$

Giaquinta & Hildebrandt, 1996

First variation (not required for this course)

$$\frac{\delta P}{\delta u} = \int_0^1 \left(v \frac{\partial F}{\partial u} + v' \frac{\partial F}{\partial u'} \right) dx \quad \text{for every } v$$

our old friend, test function !!

Weak form: $\underbrace{\int_0^1 v(x)\left(\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right)\right)dx}_{\text{integral}} + \underbrace{\left[v\frac{\partial F}{\partial u'}\right]_0^1}_{\text{boundary terms}} = 0$

Note that this is satisfied for **ALL** test functions

Euler-Lagrange equation for u:

$$\frac{\partial F}{\partial u} - \frac{d}{dx}\left(\frac{\partial F}{\partial u'}\right) = 0.$$

Example (Ex. 2.20; P. 36)

function u satisfies

$$\int_0^L u'v'\,dx + u'(0)\,v(0) + u(0)\,v'(0) + \mu u(0) v(0)$$

$$- \int_0^L f(x)\,v(x)\,dx - d_L v(L) - g_0 v'(0)$$

$$- \mu g_0 v(0) = 0.$$

for all $v \in \mathcal{V} = \{v : [0,L] \to \mathbb{R} \text{ smooth}\}$

For general procedure, See P. 35 ~ 36

Step 1: eliminate the derivative on v

$$u'(L)\,v(L) - u'(0)\,v(0) - \int_0^L u'' v\,dx + u'(0)\,v(0)$$

$$+ u(0)\,v'(0) + \mu\, u(0)\,v(0) = \int_0^L f v\,dx + d_L v(L)$$

$$+ g_0 v'(0) + \mu g_0 v(0)$$

Step 2: Collect v terms

$$\int_0^L (u'' + f)\,v\,dx = \left(u'(L) - d_L\right) v(L)$$

$$+ \left(u(0) - g_0\right) v'(0) + \mu\left(u(0) - g_0\right) v(0)$$

For $v \in \mathcal{V}$: $\rightarrow$ $v(0) = v(L) = v'(0) = 0$

implying RHS $= 0$

We conclude: $\int_0^L (u'' + f)\,v\,dx = 0$

Step 3: Obtain PDE & B.C.s

$$\rightarrow \quad u''(x) + f(x) = 0 \qquad x \in (0, L).$$

u needs to satisfy this PDE

For such u, the previous RHS should be satisfied for all v, not just $v(0) = v(L) = v'(0) = 0$ ~ $v \in \mathcal{V}$.

$$\rightsquigarrow \quad 0 = \left(u'(L) - d_L\right) v(L) + \left(u(0) - g_0\right) v'(0)$$
$$+ \mu \left(u(0) - g_0\right) v(0)$$

if $v(L) \neq 0 \Rightarrow u'(L) - d_L = 0$.

(Neumann B.C.s)

if $\left.\begin{array}{l} v'(0) \neq 0 \\ v(0) \neq 0 \end{array}\right\} \quad u(0) - g_0 = 0$

(Dirchilet B.C.s)

Euler-Lagrange equation:

$$u''(x) + f(x) = 0, \quad x \in (0, L)$$
$$u(0) = g_0,$$
$$u'(L) = d_L$$

Conceptual Clarifications

test functions

~ test how well trial functions satisfy sol'n

(Recall last P.S.)

how do we use

$\Downarrow$

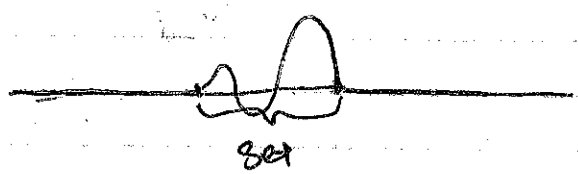
"A test function is an infinitely differentiable function of compact support." (NIST. Math. Func)

how does it look like

$\Downarrow$

"A function has compact support if it is zero outside of a compact set." (Wolfram)

(topological space)

1D example 

example $\int_0^1 u'' v \, dx < \infty$ ... test & trial func. not necessarily same.

$\Downarrow$

u must be twice differentiable

doesn't even have to be continuous

basis functions

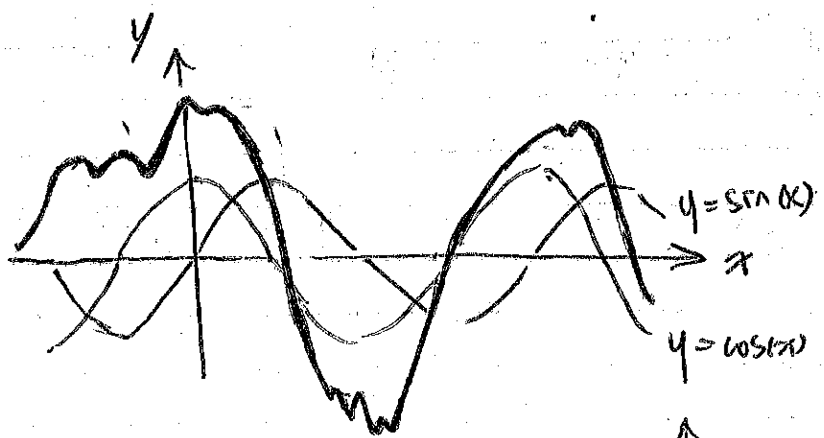
↳ usually referred in the context of **approximation** in FEA

$$u(x) = \sum_{i=1}^{n} c_i \varphi_i$$

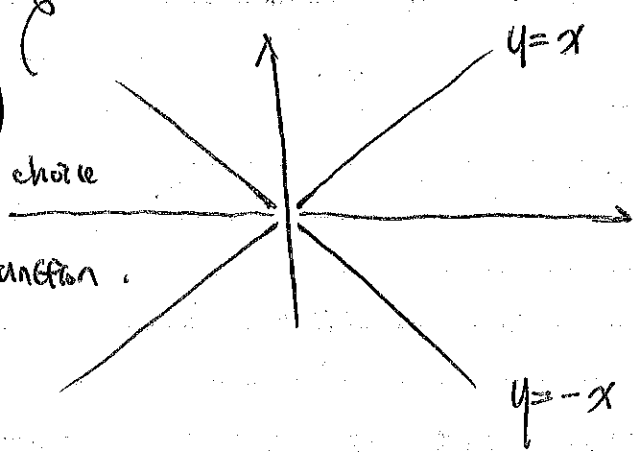
"an element of a particular basis for a function space."

↳ every function in the function space can be represented as a linear combination of basis functions.

Example



use basis function trying to interpolate a weird-shaped function

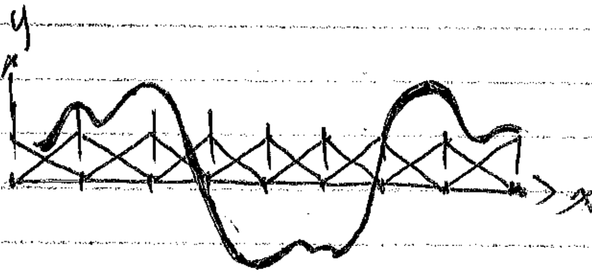


probably x a good choice of basis function.

Shape functions ----- (usually referred specifically in FEA) ———

The shape function is the function which interpolates the solution between the discrete values obtained at the mesh nodes

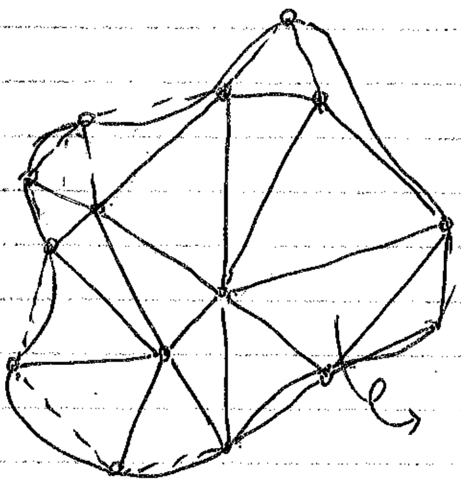
(Credit: Roberto Lacerda de Orio)



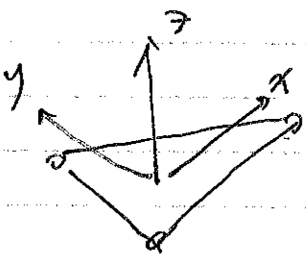
So, in FEA, sometimes they (the 3 functions) are talking about the same thing!

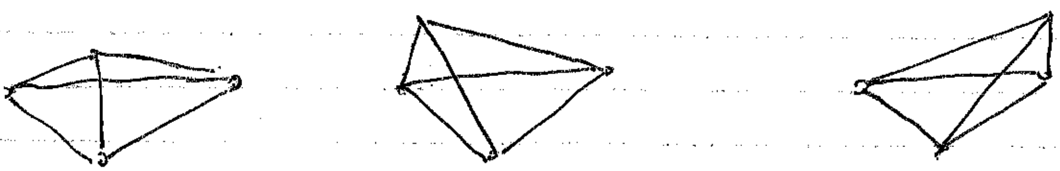
Much easier to approximate :) !!!

... in higher dimensions



pick this element





three basis functions!

Example 2.40

base space $\mathcal{V}_h = \text{span}(\{1, x, x^2, x^3\})$

test space $\mathcal{V}_h^0 = \text{span}(\{x, x^2, x^3\})$

trial space $\mathcal{S}_h = \{3 + v_h \mid v_h \in \mathcal{V}_h^0\}$

$$\begin{cases} m = 4 \\ n = 3 \end{cases} \qquad \underbrace{\underbrace{\overset{x}{N_1(x)} \quad \overset{x^2}{N_2(x)} \quad \overset{x^3}{N_3(x)}}_{n}, \quad \overset{1}{N_4(x)}}_{m}$$

$\bar{u}_h(x) = 3N_4(x) = 3.$

Recall definition of consistency

$R_h(u, v_h) = 0 \quad a_h(u_h, v_h) = l_h(v_h)$

$\Downarrow$ solves

$$a_h(u, v_h) = l_h(v_h)$$

$$\begin{cases} a_h(u_h, N_1) = l_h(N_1) \\ a_h(u_h, N_2) = l_h(N_2) \\ a_h(u_h, N_3) = l_h(N_3) \end{cases}$$

$\because u_h \in \mathcal{S}_h$, $\therefore \bar{u}_4 = 3$.

▷ load vector.

$$\underline{F} = \begin{bmatrix} l_h(N_1) \\ l_h(N_2) \\ l_h(N_3) \\ \bar{u}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

▷ Stiffness matrix

$$\underline{\underline{K}} = \begin{bmatrix} a_h(N_1, N_1) & a_h(N_2, N_1) & a_h(N_3, N_1) & a_h(N_4, N_1) \\ a_h(N_1, N_2) & a_h(N_2, N_2) & a_h(N_3, N_2) & a_h(N_4, N_2) \\ a_h(N_1, N_3) & a_h(N_2, N_3) & a_h(N_3, N_3) & a_h(N_4, N_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

write some codes (MATLAB & Python) to solve for the numerical values in K_{ij}.

Solving for $\underline{\underline{K}}\,\underline{U} = \underline{F} \rightarrow \underline{U} = \underline{\underline{K}}^{-1}\underline{F}$

$$\underline{U} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \dashrightarrow u_h(x) = u_1 N_1(x) + u_2 N_2(x) + u_3 N_3(x) + u_4 \underbrace{N_4(x)}_{1} \quad (-\bar{u}_4)$$

Some values

(from Yi Shu)

1/27/2025 Problem Session #3.

2 point BVP. $\underbrace{f(x)=1,\ g=0,\ h=0}$

constant D & μ parameters.

find smooth u s.t.

$$-Du'' + \mu u' = f \qquad x \in (0,1)$$
$$u(0) = g$$
$$u(1) = h.$$

$\Rightarrow$ exact solution: $u(x) = \frac{1}{\mu}\left(x - \frac{1-e^{\frac{\mu}{D}x}}{1-e^{\frac{\mu}{D}}}\right)$

Standard procedure

(a) form residual: $r = -Du'' + vu' - f$.

exact sol'n should satisfy $r=0$ $x \in (0,1)$

(b). for $v \in \mathcal{V}$ integrated over $(0,1)$,

$$\int_0^1 r(x)\, v(x)\, dx = 0$$

(c) integration by part. for any $v \in \mathcal{V}$.

$$\mu \int_0^1 u' v \, dx - D u' v \Big|_0^1 + D \int_0^1 u' v' \, dx$$

$$= \int_0^1 v f \, dx, \quad x \in (0, 1).$$

$\cdots (*)$.

(d). Use B.C.s & I.C.s for v, $\rightarrow$ we do not have requirement for u' @ $x=0$ & $x=1$.

Eq. $(*)$ holds
$$\begin{cases} v(0) = 0 \\ v(1) = 0 \end{cases}$$

Eq. $(*)$ becomes

$$\mu \int_0^1 u' v \, dx + D \int_0^1 u' v' \, dx = \int_0^1 v f \, dx. \quad x \in (0, 1).$$

Formulate weak form. Find $u \in \mathcal{S}$ s.t.

$a(u, v) = l(v)$ for all $v \in \mathcal{V}$.

$$a(u, v) = \mu \int_0^1 u' v \, dx + D \int_0^1 u' v' \, dx.$$

$$l(v) = \int_0^1 f v \, dx$$

$\mathcal{S} = \{u : [a,b] \to \mathbb{R},\ \text{smooth} \mid u(0) = g,$
$u(1) = h\}.$

$\mathcal{V} = \{v : [a,b] \to \mathbb{R},\ \text{smooth} \mid v(0) = 0,$
$v(1) = 0\}.$

Note that $a(u, v) \neq a(v, u)$.

→ State Galerkin formulation.

Let $\mathcal{S}_h \subset \mathcal{S}$, $\mathcal{V}_h \subset \mathcal{V}$.

Find $u_h \in \mathcal{S}_h$ s.t. ... (**)

$$a(u_h, v_h) = \ell(v_h) \quad \text{for all } v_h \in \mathcal{V}_h$$

because $g = 0$, $h = 0$. $\mathcal{S}$ & $\mathcal{V}$ are the same

Consider equidistant mesh w/ nodes $x_a = \frac{a}{N} = a\Delta x$ in $[0, 1]$. $a = 0, 1, 2, \ldots, N$. piecewise linear shape functions $\{N_a\}$ defined to span $\mathcal{S}_h$ & $\mathcal{V}_h$.

$$N_a = \begin{cases} 0, & x < x_{a-1} \\ \dfrac{x - x_{a-1}}{x_a - x_{a-1}}, & x_{a-1} \le x < x_a. \\ 1, & \text{if } x = x_a \\ \dfrac{x_{a+1} - x}{x_{a+1} - x_a} & x_a < x \le x_{a+1} \\ 0 & x_{a+1} < x \end{cases}$$

approximation writes.

$$v_h = \sum_{a=1}^{N-1} N_a v_a$$

$$u_h = \sum_{b=1}^{N-1} N_b u_b$$

proceed to compute $a(u_h, v_h)$ & $l(v_h)$.

$$a(v_h, u_h) = a\left(\sum_{a=1}^{N-1} N_a v_a, \sum_{b=1}^{N-1} N_b u_b\right).$$

$$= \sum_{a=1}^{N-1} \sum_{b=1}^{N-1} v_a u_b \, a(N_a, N_b)$$

$$l(v_h) = l\left(\sum_{a=1}^{N-1} N_a v_a\right)$$

$$= \sum_{a=1}^{N-1} v_a l(N_a)$$

rewriting Eq. (**)

$$\sum_{a=1}^{N-1} \sum_{b=1}^{N-1} v_a u_b \, a(N_b, N_a) = \sum_{a=1}^{N-1} v_a \, l(N_a) \quad \text{for all } v_a$$

$$\sum_{b=1}^{N-1} a(N_b, N_a) = l(N_a)$$

entries of load vector $\underline{F}$, $f_a = l(N_a)$.

stiffness matrix $\underline{\underline{K}}$, $K_{ab} = a(N_b, N_a)$

one can solve for $\underline{U}$

$$\underline{\underline{K}}\,\underline{U} = \underline{F}$$

$$a(N_b, N_a) = \mu \int_0^1 N_a' N_b \, dx + D \int_0^1 N_b' N_a' \, dx$$

$$l(N_a) = \int_0^1 N_a \, dx$$

Case study $N=3$. nodes in the mesh

$0, \frac{1}{3}, \frac{2}{3}, 1.$

$\underline{\underline{K}} \to 2\times 2$

$\Downarrow$

$\Delta x = \frac{1}{3}$.

$\underline{U} \to 2\times 1$

$\underline{F} \to 2\times 1$

$$K_{11} = a(N_1, N_1) = \mu \int_0^1 N_1 N_1' dx + D\int_0^1 N_1' N_1' dx$$

$$= \frac{2D}{\Delta x}$$

$$K_{22} = a(N_2, N_2) = \mu \int_0^1 N_2 N_2' dx + D\int_0^1 N_2' N_2' dx$$

$$= \frac{2D}{\Delta x}.$$

$$K_{12} = a(N_2, N_1) = \mu \int_0^1 N_2' N_1 dx + D\int_0^1 N_2' N_1' dx$$

$$= \mu/2 - D/\Delta x$$

$$K_{21} = a(N_1, N_2) = \mu \int_0^1 N_1' N_2 dx + D\int_0^1 N_1' N_2' dx$$

$$= -\mu/2 - D/\Delta x$$

Solving for load vector.

$$\underline{F} \rightarrow \begin{cases} F_1 = 1 \\ F_2 = 1. \end{cases}$$

linear system.

$$\begin{bmatrix} 2D/\Delta x & \mu/2 - D/\Delta x \\ -\mu/2 - D/\Delta x & 2D/\Delta x \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\det(\underline{\underline{K}}) = 3D^2/\Delta x^2 + \mu^2/4 \neq 0$$

$$\Downarrow$$

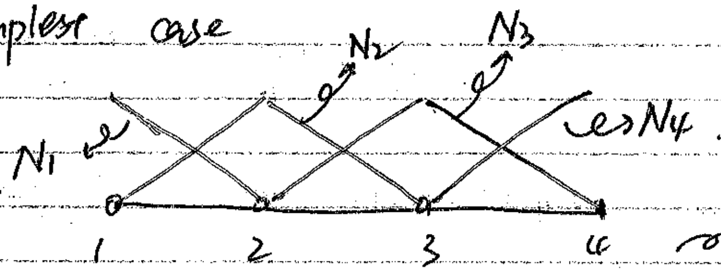
the system is invertible.

Problem Session 4 2/3/2025

- Local-to-Global Map
- Examples on FEM implementation
- Q & A.

Local to Global Map

Simplest case



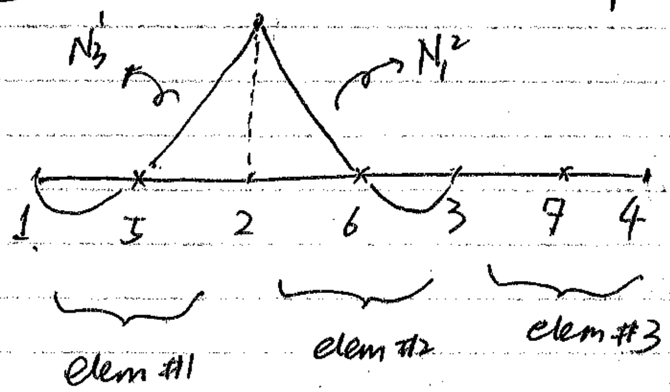
node number

$$LG = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

element #1, elem. #2, elem. #3.

$$LG = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 2 & 3 & 4 \end{bmatrix}$$

Example 3.11 continuous piecewise quadratic functions.



$N_1 = N_1^1$

$N_2 = N_3^1 \,\&\, N_1^2$

$N_3 = N_3^2 \,\&\, N_1^3$

$N_4 = N_3^3$

$N_5 = N_2^1$

$N_6 = N_2^2$, $N_7 = N_2^3$

N_a^b → b: element number
→ a: shape functions (local)

Modified example from Philip DePond

▷ Consider a 1D diffusion-advection equation.

given constant $k < 0$, f, v.

... find T smooth enough s.t.

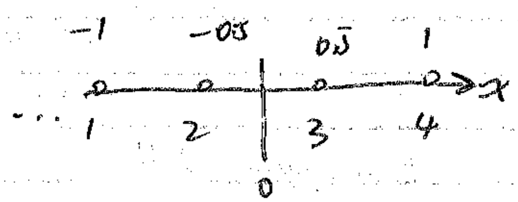
$$k \frac{d^2 T}{dx^2} + v \frac{dT}{dx} = f \quad : \quad \text{in } \Omega \in [-1, 1].$$

$$T(x=-1) = T_1$$

$$T(x=1) = T_2 \qquad \text{B.C.s.}$$

Consider a simple mesh w/ 4 nodes using linear elements

Node	coordinate
1	-1
2	-0.5
3	0.5
4	1



State Galerkin form:

Starting from strong form:

$$\int_\Omega \left(k \frac{d^2 T}{dx^2} + v \frac{dT}{dx} \right) w \, dx = \int_\Omega f w \, dx$$

$$\int T'' w\,dx = \int w\,dT'$$

$$\hookrightarrow T'w\Big|_{\partial\Omega} - \int T'w'dx$$

$$-k\int_\Omega \frac{dT}{dx}\frac{dw}{dx}dx + v\int_\Omega \frac{dT}{dx}w\,dx = \int_\Omega f w\,dx$$

the Galerkin form is stated:

$$a(w, T) = \int_\Omega \left(k\frac{dw}{dx}\frac{dT}{dx} - v\frac{dT}{dx}w\right)dx$$

$$l(w) = -\int_\Omega f w\,dx$$

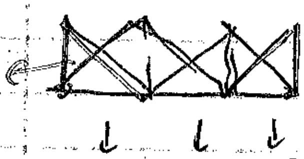
Find $T_h \in \mathcal{T}_h = \text{span}\{N_1, N_2, N_3, N_4\}$ s.t.

$$a(w_h, T_h) = l(w_h) - a(w_h, T_h^g),$$

$$\forall w_h \in \mathcal{W}_h = \mathcal{T}_h$$

~ Determine the LG matrix

Impose B.C.s



$$LG = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

#1 #2 #3

Global statement for the finite element problem

Dirichlet B.C.s: $T_h^g = T_1 N_1 + T_4 N_4$.

unknown part: $T_h = T_2 N_2 + T_3 N_3$

Full soln: $T_h^{total} = T_h^g + T_h = \sum_{i=1}^{4} T_i N_i$

test function: $w_h = \sum_{i=1}^{4} w_i N_i$

$w_1 = w_4 = 0$ (due to Dirichlet B.Cs)

Recall def'n of bilinear func.

$$a(u, w+v) = a(u,w) + a(u,v)$$

...

Substitute into bilinear form:

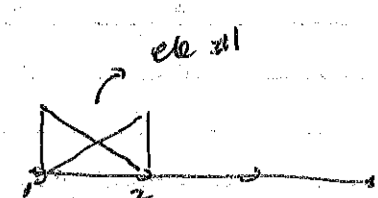
$$\sum_{i=1}^{4} \sum_{j=1}^{4} w_i \, a(N_i, N_j) T_j = \sum_{i=1}^{4} w_i \, l(N_i) - \sum_{i=1}^{4} w_i \, a(N_i, T_h^g)$$

Since $w_1 = w_4 = 0$, system reduces to

$$\sum_{j=2}^{3} a(N_i, N_j) T_j = l(N_i) - a(N_i, T_h^g)$$

$i = 2, 3$.

Local version of Finite Element.



Element #1

$$K_{ab}^1 = \int_{\Omega^1} \left(k \frac{dN_a^1}{dx} \frac{dN_b^1}{dx} - v \frac{dN_b^1}{dx} N_a^1 \right) dx, \quad a, b = 1, 2$$

$$F_a^1 = -\int_{\Omega^1} f N_a^1 \, dx - a(N_a^1, T_h^g), \quad a = 1, 2.$$

K^1, F^1 corresponding to $LG(a, 1)$ & $LG(b, 1)$

Element #2 (nodes #2 & #3)

$$K^2_{ab} = \int_{\Omega^2} \left(k \frac{dN_a^2}{dx} \frac{dN_b^2}{dx} - v \frac{dN_b^2}{dx} N_a^2 \right) dx, \quad a,b = 1,2.$$

$$F_a^2 = -\int_{\Omega^2} f N_a^2 \, dx - a(N_a^2, T_h^g), \quad a = 1,2.$$

K^2, F^2 correspond to $LG(a,2)$ & $LG(b,2)$

... same procedure with elements #3 & #4.

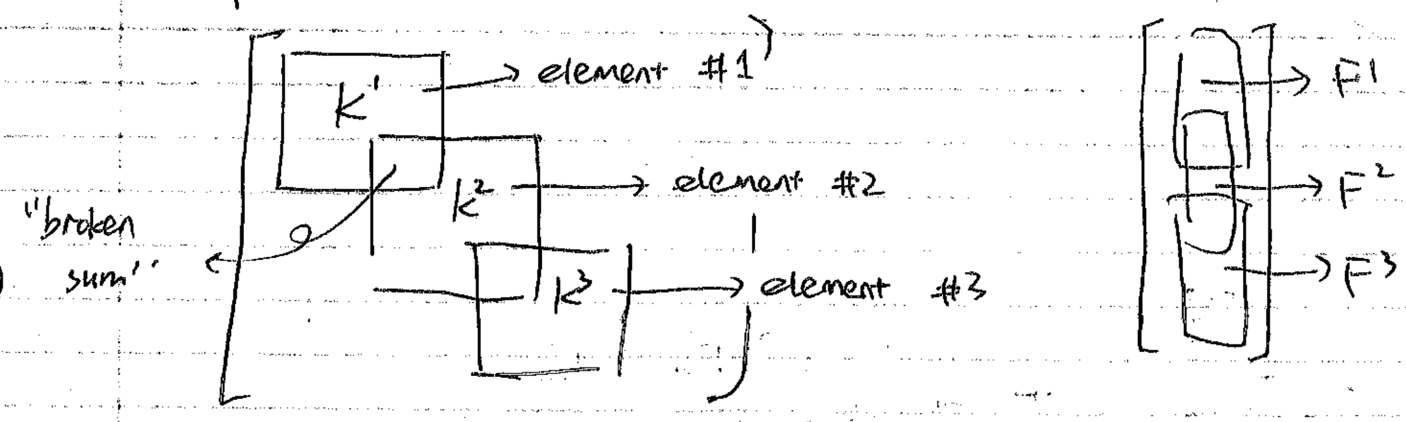
↳ corresponding to $LG(a,3)$, $LG(b,3)$ ...?
$LG(a,4)$, $LG(b,4)$

(Important !!) Assemble the Global System.

$$K_{LG(a,e),\,LG(b,e)} \leftarrow K_{LG(a,e),\,LG(b,e)} + K^e_{ab} \quad \text{for all } e, a, b$$

$$F_{LG(a,e)} \leftarrow F_{LG(a,e)} + F^e_a$$

Stiffness matrix



Final step: Solve the global system

$$KT = F$$

$$T = K^{-1}F$$

after solving for T, the sol'n:

$$T_h = T_1 N_1 + T_2 N_2 + T_3 N_3 + T_4 N_4$$

... implement these in

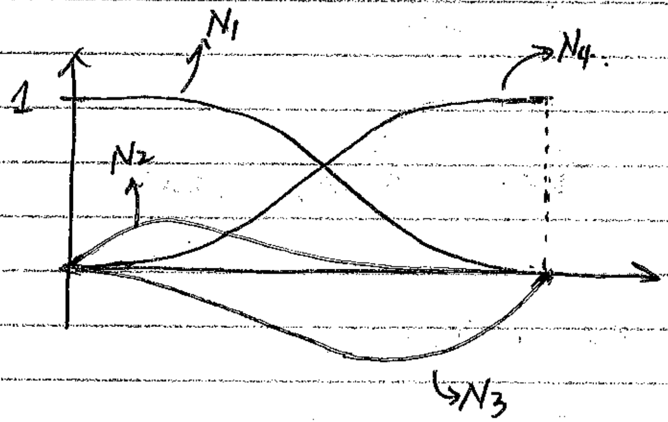
Python / MATLAB

~ your first FEM code !

Problem Session #5.   2/10/2025.

... LG matrix.

Definitions of Hermite element



Eqns. (4.79)

$$N_1^e(x) = \left(\frac{x_2^e - x_1}{x_2^e - x_1^e}\right)^2 \left(1 + 2\frac{x - x_1^e}{x_2^e - x_1^e}\right)$$

$$N_2^e(x) = \left(\frac{x_2^e - x}{x_2^e - x_1^e}\right)^2 (x - x_1^e)$$

$$N_3^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 \left(1 + 2\frac{x - x_2^e}{x_1^e - x_2^e}\right)$$

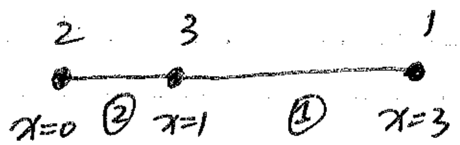
$$N_4^e(x) = \left(\frac{x_1^e - x}{x_1^e - x_2^e}\right)^2 (x - x_2^e)$$

Cubic polynomial in e :

$$f^e(x) = \phi_1^e N_1^e(x) + \phi_2^e N_2^e(x) + \phi_3^e N_3^e(x) + \phi_4^e N_4^e(x)$$

Example 4.8

Consider a two-element mesh.



nodal coordinates: $x_1 = 3, \quad x_2 = 0, \quad x_3 = 1$

Local-to-global map writes:

$$LG = \begin{bmatrix} 5 & 3 \\ 6 & 4 \\ 1 & 5 \\ 2 & 6 \end{bmatrix}$$

Using the definition of LG matrix:

$$N_A = \sum_{\{(a,e) \mid LG(a,e) = A\}} N_a^e$$

One writes for $A = 1, \ldots, 6$

$$N_1 = N_3^1$$

$$N_2 = N_4^1$$

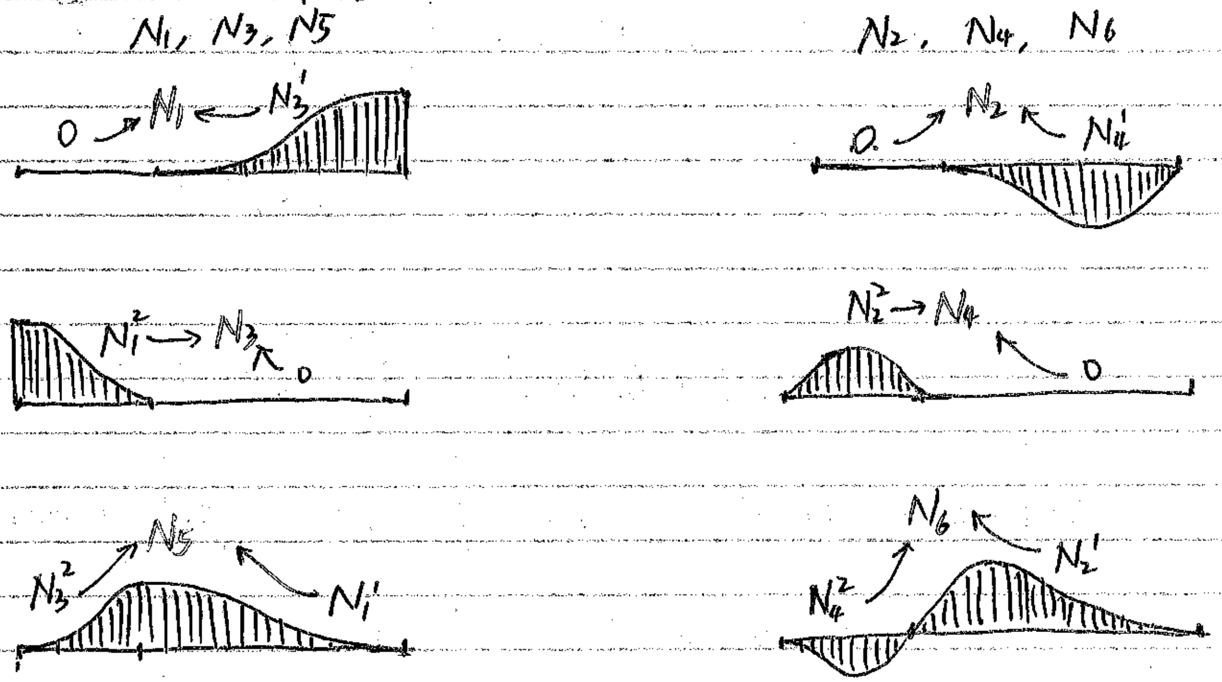
$$N_3 = N_1^2$$

$$N_4 = N_2^2$$

$$N_5 = N_1^1 + N_3^2$$

$$N_6 = N_2^1 + N_4^2$$

Global shape functions



... just a review

Recall the standard form for a 2nd order diff. eqn.

$$-\left(k(x)\,u'(x)\right)' + b(x)\,u'(x) + c(x)\,u(x) = f(x) \qquad \cdots (2.1)$$

After variational formulation, some algebra, defining the finite element, ..., we have:

$$a_h(u_h, v_h) = \sum_{e=1}^{n_{el}} \underbrace{\int_{K_e} \left[k(x)\,u_h'(x)\,v_h'(x) + b(x)\,u_h'(x)\,v_h(x) + c(x)\,u_h(x)\,v_h(x) \right] dx}_{= a_h^e(u_h, v_h)}$$

... assembly step

$$= \sum_{e=1}^{n_{el}} a_h^e(u_h, v_h)$$

$$l_h(v_h) = k(L)\, d_L v_h(L) + \sum_{e=1}^{n_{el}} \underbrace{\int_{K_e} f(x)\, v_h(x)\, dx}_{l_h^e(v_h)}$$

$$= k(L)\, d_L v_h(L) + \sum_{e=1}^{n_{el}} l_h^e(v_h)$$

Consider BVP: constant f, EI, find smooth u s.t.

$$(EI\, u_{,xx})_{,xx} = f \qquad x \in (0,1)$$

$$u(0) = 0$$
$$u'(0) = 0$$
$$u(1) = 0$$
$$u'(1) = 0$$

Galerkin form:

Find $u^h \in \mathcal{S}_h \subset \mathcal{S} = \{u : [0,1] \to \mathbb{R} \text{ smooth} \mid u(0)=0,\ u'(0)=0,\ u(1)=0,\ u'(1)=0\}$

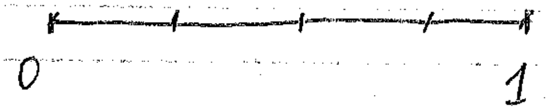
$$a(u_h, v_h) = l(v_h)$$

$$a(u_h, v_h) = \int_0^1 u_h''\, EI\, v_h''\, dx$$

$$l(v_h) = \int_0^1 f v_h\, dx.$$

for all $v_h \in \mathcal{V}_h \subset \mathcal{V} = \{v : [0,1] \to \mathbb{R} \text{ smooth} \mid v(1)=0,\ v'(1)=0,\ v(0)=0,\ v'(0)=0\}$

consider a mesh of 4 elements l_1, l_2, l_3, l_4



$l_e = x_2 - x_1 > 0$

$\Omega^e = [x_1, x_2]$

$$N_1^e = \frac{-(x - x_2)^2\left[-l_e + 2(x_1 - x)\right]}{l_e^3}$$

$$N_2^e = \frac{(x - x_1)(x - x_2)^2}{l_e^2}$$

$$N_3^e = \frac{(x - x_1)^2\left[l_e + 2(x_2 - x)\right]}{l_e^3}$$

$$N_4^e = \frac{(x - x_1)^2(x - x_2)}{l_e^2}$$

for general element w/ length l_e,

entries $k_{ab}^e = a(N_b^e, N_a^e)$

$\rightarrow$ take second derivative of N_1^e:

$$N_{1,xx}^e = \frac{2(l_e + 6x - 2x_1 - 4x_2)}{l_e^3} = \frac{2(-3l_e + 6x - 6x_1)}{l_e^3}$$

using change of variable: $x = x_1 + \xi(x_2 - x_1)$, $\xi \in [0, 1]$

we have: $dx = l_e \, d\xi$

$$\frac{d^2x}{d\xi^2} = 0$$

$$\frac{dN}{dx} = \frac{dN}{d\xi}\frac{d\xi}{dx}.$$

$$\frac{d^2N}{dx^2} = \frac{d^2N}{d\xi^2}\left(\frac{d\xi}{dx}\right)^2 + \frac{dN}{d\xi}\frac{d^2\xi}{dx^2}$$

because $\frac{d^2\xi}{dx^2} = 0.$

$$\frac{d^2N}{dx^2} = \frac{d^2N}{d\xi^2}\left(\frac{d\xi}{dx}\right)^2$$

We have $N_1^e = (1-\xi)^2(1+2\xi)$

$$\frac{d^2N_1^e}{d\xi^2} = -6+12\xi$$

$$\frac{d^2N_1^e}{dx^2} = \frac{d^2N_1^e}{d\xi}\left(\frac{d\xi}{dx}\right)^2 = \frac{-6+12\xi}{l_e^2}$$

We can calculate k_1^e as an example:

$$a(N_1^e, N_1^e) = EI\int_{x_1}^{x_2} N_{1,xx}^e N_{1,xx}^e \, dx$$

$$= \frac{36EI}{l_e^3}\int_0^1 (-1+2\xi)^2 \, d\xi.$$

$$= \cdots = \frac{12EI}{l_e^3}$$

Using this transformation, we derive the other shape functions $N_1^e = (1-\xi)^2(1+2\xi)$

$$N_2^e = l_e \xi (\xi - 1)^2$$

$$N_3^e = \xi^2 (3 - 2\xi)$$

$$N_4^e = l_e \xi^2 (\xi - 1)$$

2nd order derivative:

$$N_{1,xx}^e = -6 + 12\xi$$

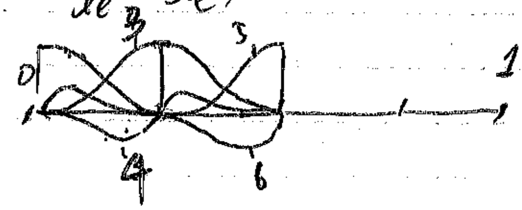
$$N_{2,xx}^e = l_e (6\xi - 4)$$

$$N_{3,xx}^e = 6 - 12\xi$$

$$N_{4,xx}^e = l_e (6\xi - 2)$$

elemental stiffness matrix.

$$\underline{\underline{k}}^e = EI \begin{bmatrix} \frac{12}{l_e^3} & \frac{6}{l_e^2} & -\frac{12}{l_e^3} & \frac{6}{l_e^2} \\ \frac{6}{l_e^2} & \frac{4}{l_e} & -\frac{6}{l_e^2} & \frac{2}{l_e} \\ -\frac{12}{l_e^3} & -\frac{6}{l_e^2} & \frac{12}{l_e^3} & -\frac{6}{l_e^2} \\ \frac{6}{l_e^2} & \frac{2}{l_e} & -\frac{6}{l_e^2} & \frac{4}{l_e} \end{bmatrix}$$



Write LG matrix

$$LG = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 3 & 5 & 7 & 9 \\ 4 & 6 & 8 & 10 \end{bmatrix}$$

Assemble the global stiffness matrix

$$K = EI\begin{bmatrix} \frac{12}{l_1^3} & -\frac{6}{l_1^2} & & & & \\ -\frac{6}{l_1^2} & \frac{4}{l_1} & & & & \\ & & & & & \\ & & & & \frac{12}{l_4^3} & \frac{6}{l_4^2} \\ & & & & \frac{6}{l_4^2} & \frac{4}{l_4} \end{bmatrix} +$$

$$EI\begin{bmatrix} \frac{12}{l_2^3} & \frac{6}{l_2^2} & -\frac{12}{l_2^3} & \frac{6}{l_2^2} & 0 & 0 \\ \frac{6}{l_2^2} & \frac{4}{l_2} & -\frac{6}{l_2^2} & \frac{2}{l_2} & 0 & 0 \\ -\frac{12}{l_2^3} & -\frac{6}{l_2^2} & \frac{12}{l_2^3}+\frac{12}{l_3^3} & -\frac{6}{l_2^2}+\frac{6}{l_3^2} & -\frac{12}{l_3^3} & \frac{6}{l_3^2} \\ \frac{6}{l_2^2} & \frac{2}{l_2} & -\frac{6}{l_2^2}+\frac{6}{l_3^2} & \frac{4}{l_2}+\frac{4}{l_3} & -\frac{6}{l_3^2} & \frac{2}{l_3} \\ 0 & 0 & -\frac{12}{l_3^3} & -\frac{6}{l_3^2} & \frac{12}{l_3^2} & -\frac{6}{l_3^2} \\ 0 & 0 & \frac{6}{l_3^2} & \frac{2}{l_3} & -\frac{6}{l_3^2} & \frac{4}{l_3} \end{bmatrix}$$

Assemble force vector $f_a^e = f\int_{x_1}^{x_2} N_a^e\,dx$

$f_1^e = l_e f\int_0^1 (1-\xi)^2(1+2\xi)\,d\xi = \frac{1}{2} f l_e$

$f_2^e = l_e^2 f\int_0^1 \xi(\xi-1)^2\,d\xi = \frac{1}{12} f l_e^2$

$f_3^e = l_e f\int_0^1 \xi(3-2\xi)\,d\xi = \frac{1}{2} f l_e$

$f_4^e = l_e^2 f\int_0^1 \xi^2(\xi-1)\,d\xi = -\frac{1}{12} f l_e^2$

Assemble global $\underline{F}$

$$\underline{F} = f \begin{bmatrix} \frac{1}{2}l_1 + \frac{1}{2}l_2 \\ -\frac{1}{12}l_1^2 + \frac{1}{12}l_2^2 \\ \frac{1}{2}l_2 + \frac{1}{2}l_3 \\ -\frac{1}{12}l_2^2 + \frac{1}{12}l_3^2 \\ \frac{1}{2}l_3 + \frac{1}{2}l_4 \\ -\frac{1}{12}l_3^2 + \frac{1}{12}l_4^2 \end{bmatrix}$$

Problem Session #6 2/15/2025

⊳ 2D !!!

Recall Problem Session #4, we solved 1D diffusion-advection equation using FEA. Today we are going to solve it in 2D.

2D diffusion-advection problem, $k<0$, f, v_1 & v_2 are constants.

find T smooth enough such that.

$$k\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right)+v_1\frac{\partial T}{\partial x}+v_2\frac{\partial T}{\partial y}=f \quad \text{in } \Omega=[-1,1]\times[0,1].$$

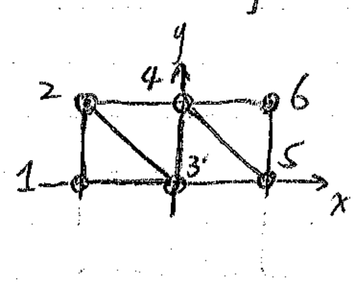
$$T(-1,y)=\tilde{T}_1$$

$$T(1,y)=\tilde{T}_2$$

$$T_{,i}\,n_i=0 \quad \text{on } y=-1 \text{ and } y=1.$$

Consider a simple mesh → 6 nodes using linear triangles.

node	coordinate
# 1	(-1, 0)
# 2	(-1, 1)
# 3	(0, 0)
# 4	(0, 1)
# 5	(1, 0)
# 6	(1, 1)



State the Galerkin formulation:

$\rightarrow \; T_i w_i = T_1 w_1 + T_2 w_2$

$$\int_{\Omega} (k T_{,ii} + v_i T_{,i}) w \, d\Omega = \int_{\Omega} f w \, d\Omega$$

$$\int_{\Omega} (-k T_{,i} w_{,i} + v_i T_{,i} w) \, d\Omega + \int_{\partial\Omega} k w T_{,i} n_i \, d\Gamma$$

$$= \int_{\Omega} f w \, d\Omega .$$

Since $w = 0$ on Γ_g, therefore $\int_{\Gamma_g} k w T_{,i} n_i \, d\Gamma = 0.$

$$\int_{\Omega} (-k T_{,i} w_{,i} + v_i T_{,i} w) \, d\Omega + \int_{\Gamma_h} k w T_{,i} n_i \, d\Gamma$$

$$= \int_{\Omega} f w \, d\Omega$$

$\because \; T_{,i} n_i = 0$ on Γ_h, we then have

$$\int_{\Omega} (k T_{,i} w_{,i} - v_i T_{,i} w) \, d\Omega = -\int_{\Omega} f w \, d\Omega$$

$$a(T, w) = \int_{\Omega} (k T_{,i} w_{,i} - v_i T_{,i} w) \, d\Omega$$

$$l(w) = -\int_{\Omega} f w \, d\Omega$$

The Galerkin formulation is stated as:

Find $T_h \in \mathcal{T}_h = \text{span}\{N_3, N_4\}$ s.t.

$$a(T_h, w_h) = l(w_h) - a(w_h, T_h^g),$$
$$\forall w_h \in \mathcal{W}_h = \mathcal{T}_h$$

* LG matrix:

$$LG = \begin{bmatrix} 1 & 2 & 4 & 4 \\ 3 & 3 & 3 & 5 \\ 2 & 4 & 5 & 6 \end{bmatrix}$$

Global version of finite element.

$$T_h^g = T_1 N_1 + T_2 N_2 + T_5 N_5 + T_6 N_6.$$

$$\left.\begin{aligned} T_h^a &= \sum_{b=3}^{4} T_b N_b \end{aligned}\right\} T_h = T_h^a + T_h^g$$

$$w_h = \sum_{a=3}^{4} W_a N_a.$$

Substitute into the weak form.

$$a\left(\sum_{b=3}^{4} T_b N_b, \sum_{a=3}^{4} T_a N_b\right) = l\left(\sum_{a=3}^{4} W_a N_a\right) - a\left(T_h^g, \sum_{a=3}^{4} W_a N_a\right).$$
$$\forall w_h = \mathcal{W}_h = \mathcal{T}_h.$$

reorganize the sign of summation.

$$\sum_{a=3}^{4}\sum_{b=3}^{4} w_a\, a(T_b N_b, N_a) = \sum_{a=3}^{4} w_a\, l(N_a) - \sum_{a=3}^{4} w_a\, a(T_h^g, N_a), \quad \forall w_h \in \mathcal{V}_h = \mathcal{T}_h$$

We can reformulate the equation as.

$$\sum_{b=3}^{4} a(N_b, N_a) T_b = l(N_a) - a(T_h^g, N_a).$$

$$K_{LG(b),LG(a)} = a(N_b, N_a)$$

$$F_{LG(a)} = l(N_a) - a(T_h^g, N_a)$$

$K \rightarrow$ not symmetric because $a(T, w)$ is not symmetric.

∴ local version of finite element.

$$a(T_h, w_h)_{\Omega} = \sum_{LG}^{e} a(T_h, w_h)_{\Omega^e}$$

a simplified symbol for global assembly

$$l(w_h)_{\Omega} = \sum_{LG}^{e} l(w_h)_{\Omega^e}$$

on Ω^e $\quad T_h = \sum_{b=1}^{3} T_b^e N_b^e$

$$w^{h} = \sum_{a=1}^{3} w_a^e N_a^e$$

We have $K^e_{ab} = a(N^e_b, N^e_a)_{\Omega^e}$

$$l^e_a = l(N^e_a)_{\Omega^e} - a(T^g_h, N^e_a)_{\Omega^e}$$

$$\rightarrow a(T^g_h, N^e_a)_{\Omega^e} = K^e_{ab}\, g^e$$

$$IG^e = \begin{bmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

on Ω^1:

$$T^g_h = \tilde{T}_1 N_1' + \tilde{T}_1 N_3'$$

$$a(T^g_h, N_a')_{\Omega'} = a(\tilde{T}_1 N_1' + \tilde{T}_1 N_3', N_a')_{\Omega'}$$

$$= a(\tilde{T}_1 N_1' + 0 N_2' + \tilde{T}_1 N_3', N_a')_{\Omega'}$$

$$a(T^g_h, N_a')_{\Omega'} = a(N_1', N_a')_{\Omega'} \tilde{T}_1 + a(N_2', N_a')_{\Omega'} \cdot 0 + a(N_3', N_a')_{\Omega'} \tilde{T}_1$$

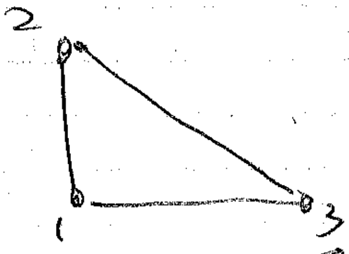
Therefore

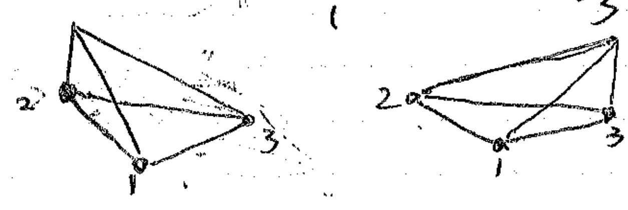
$$\begin{bmatrix} a(T^g_h, N_1')_{\Omega'} \\ a(T^g_h, N_2')_{\Omega'} \\ a(T^g_h, N_3')_{\Omega'} \end{bmatrix} = \begin{bmatrix} a(N_1', N_1') & a(N_2', N_1') & a(N_3', N_1') \\ a(N_1', N_2') & a(N_2', N_2') & a(N_3', N_2') \\ a(N_1', N_3') & a(N_2', N_3') & a(N_3', N_3') \end{bmatrix} \begin{bmatrix} \tilde{T}_1 \\ 0 \\ \tilde{T}_1 \end{bmatrix}$$

Compute $a(N_2', N_3')\big|_{\Omega'}$

$$N_3'(x, y) = x + 1$$

$$N_2'(x, y) = y$$





$$a(N_2^1, N_3^1)\big|_{\Omega^1} = \int_{\Omega^1} \left(k N^1_{3,x} N^1_{2,x} + k N^1_{3,y} N^1_{2,y} - v_1 N^1_{2,x} N^1_3 - v_2 N^1_{2,y} N^1_3\right) d\Omega .$$

$$= \int_{\Omega^1} \left(-v_1 N^1_{2,x} N^1_3\right) d\Omega$$

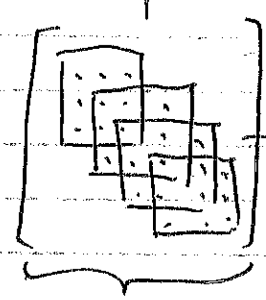
$$= -v_1 \int_{\Omega^1} y \, d\Omega = -\frac{v_1}{6} .$$

We can then do the assembly of K and F

$$K = \begin{bmatrix} K^1_{22} + K^2_{22} & K^2_{23} \\ & \\ K^2_{32} & K^2_{33} \end{bmatrix} \overset{\circ}{+} \cdots$$

the nest steps should be the same

... Dimension of overall K ?



$\rightarrow 6$

6

$$\begin{bmatrix} \vec{F} \end{bmatrix} \rightarrow 6$$

Tutorial on FEniCS: solving 2D Poisson equation

Hanfeng Zhai
hzhai@stanford.edu

January 4, 2025

Introduction

This tutorial demonstrates how to solve the Poisson equation using the finite element method (FEM) with the FEniCS library[1]. The Poisson equation is a widely used partial differential equation (PDE) that models physical phenomena such as heat conduction, electrostatics, and diffusion.

Mathematical Formulation

The Poisson equation in two dimensions is written as:

$$-\nabla \cdot (k\nabla u) = f \quad \text{in } \Omega, \tag{1}$$

where:

- u is the unknown scalar field (e.g., temperature).
- k is the thermal conductivity (assumed constant in this example).
- f is the source term (e.g., heat generation). $f = 0$ in this example.
- Ω is the computational domain ($\mathbb{R}^2$).

The boundary conditions are defined as:

$$u = g_D \quad \text{on } \Gamma_D, \tag{2}$$

$$-k\frac{\partial u}{\partial n} = g_N \quad \text{on } \Gamma_N, \tag{3}$$

where Γ_D and Γ_N are Dirichlet and Neumann boundaries, respectively, and n is the outward normal vector.

In this example, we solve Equation (1) with Dirichlet boundary conditions on all boundaries.

Implementation in FEniCS

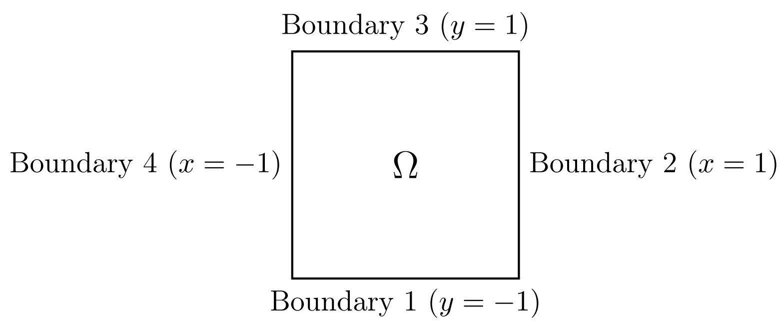
The following Python code implements the solution of the Poisson equation using FEniCS. The computational domain is a square, and the Dirichlet boundary conditions set $u = 1000$ on all edges of the domain. The source term is constant, $f = 0.2$, and $k = 2 \times 10^{-4}$. The solution is visualized as a heatmap and as a 3D surface plot.

[1] We use FEniCS 2019 version

Schematic of the Domain

The computational domain is a square defined as $[-1,1] \times [-1,1]$. The boundaries are labeled as follows:

- Boundary 1: $y = -1$,
- Boundary 2: $x = 1$,
- Boundary 3: $y = 1$,
- Boundary 4: $x = -1$.



Key Steps in the Code

1. **Mesh Generation:** The domain is discretized using a triangular mesh generated by Gmsh and converted for use in FEniCS.
2. **Boundary Conditions:** Dirichlet boundary conditions are applied on all edges of the square.
3. **Variational Formulation:** The weak form of the Poisson equation is derived as:
$$\int_\Omega k \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x, \tag{4}$$
where v is the test function.
4. **Solution Computation:** The linear system resulting from the discretization is solved, yielding the scalar field u.
5. **Visualization:** The solution is plotted as a 2D heatmap and a 3D surface.

[ ]:
```
# installing required packages
try:
    import dolfin
except ImportError:
    !wget "https://fem-on-colab.github.io/releases/fenics-install-real.sh" -O "/tmp/fenics-install.sh" && bash "/tmp/fenics-install.sh"
    import dolfin
!pip install meshio
!apt-get install gmsh
!gmsh --version
!pip install --upgrade gmsh
```

We begin with importing the necessary packages.

[ ]:
```
import gmsh, meshio
from fenics import *
import matplotlib.pyplot as plt
from matplotlib.tri import Triangulation
from dolfin import *
import numpy as np
from mesh_converter import msh_to_xdmf
```

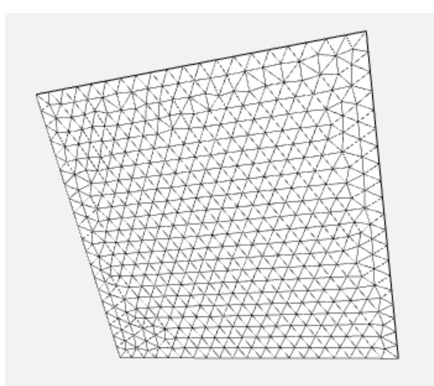
The mesh is defined accordingly given the geometry.

[ ]:
```
def Tutorialmesh(Hmax, elementOrder, elementType):
    # Given Hmax, construct a mesh to be read by FEniCS
    gmsh.initialize()
    gmsh.model.add('Tutorialmesh')
    meshObject = gmsh.model

    # Points for the outer boundary
    point1 = meshObject.geo.addPoint(-1,-1,0,Hmax, 1)
    point2 = meshObject.geo.addPoint(1,-1,0,Hmax, 2)
    point3 = meshObject.geo.addPoint(1,1,0,Hmax, 3)
    point4 = meshObject.geo.addPoint(-1,1,0,Hmax, 4)

    # Construct lines from points
    line1 = meshObject.geo.addLine(1, 2, 101)
    line2 = meshObject.geo.addLine(2, 3, 102)
    line3 = meshObject.geo.addLine(3, 4, 103)
    line4 = meshObject.geo.addLine(4, 1, 104)

    # Construct closed curve loops
    outerBoundary = meshObject.geo.addCurveLoop([line1, line2, line3, line4],␣
↪201)

    # Define the domain as a 2D plane surface with holes
    domain2D = meshObject.geo.addPlaneSurface([outerBoundary], 301)

    # Synchronize gmsh
    meshObject.geo.synchronize()

    # Add physical groups for firedrake
    meshObject.addPhysicalGroup(2, [301], name='domain')

    meshObject.addPhysicalGroup(1, [line2], 1)
    meshObject.addPhysicalGroup(1, [line3], 2)
    meshObject.addPhysicalGroup(1, [line4], 3)
    meshObject.addPhysicalGroup(1, [line1], 4)
```

```python
    # Set element order
    meshObject.mesh.setOrder(elementOrder)

    if elementType == 2:
      # Generate quad mesh from triangles by recombination
      meshObject.mesh.setRecombine(2, domain2D)

    # Generate the mesh
    gmsh.model.mesh.generate(2)
    gmsh.write('mesh.msh')

    gmsh.finalize()
    mesh_from_gmsh = meshio.read("mesh.msh")

    triangle_mesh = meshio.Mesh(
        points=mesh_from_gmsh.points,
        cells={"triangle": mesh_from_gmsh.get_cells_type("triangle")},
    )

    meshio.write("mesh.xml", triangle_mesh)
    msh_to_xdmf('mesh')
    mesh = Mesh("mesh.xml")

    return mesh
```

The mesh of the domain is generated via

[ ]:
```python
# Generate the mesh
elementOrder = 1 # Polynomial order in each element (integer)
elementType = 1  # 1 - Triangle; 2 - Quad
HMax = 0.1

mesh = Tutorialmesh(HMax, elementOrder, elementType)
'''visualize the mesh (the object)'''
mesh
```



[ ]: <dolfin.cpp.mesh.Mesh at 0x7fd76fd93f10>

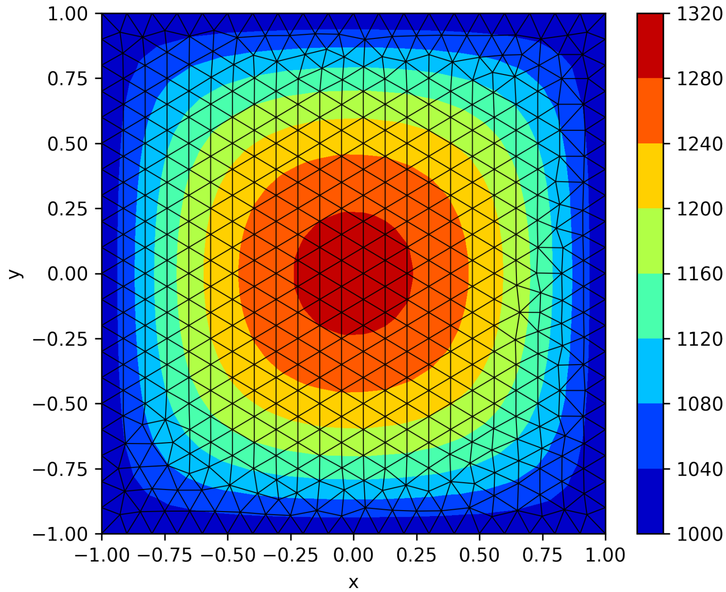
[ ]:
```
mesh = Mesh()
with XDMFFile('mesh' + ".xdmf") as infile:
    infile.read(mesh)
# load boundary markers from facet file
mvc_facet = MeshValueCollection("size_t", mesh, mesh.topology().dim() - 1)
with XDMFFile('mesh' + "_facets.xdmf") as infile:
    infile.read(mvc_facet, "facet_marker")
boundaries = cpp.mesh.MeshFunctionSizet(mesh, mvc_facet)
ds = Measure("ds", subdomain_data=boundaries)
```

[ ]:
```
V = FunctionSpace(mesh, "CG", 1)
boundary_markers = MeshFunction("size_t", mesh, mesh.topology().dim()-1)

DirBC = [DirichletBC(V, Constant(1000.0), boundaries, marker) for marker in␣
↪[1,2,3,4]]
bcs = DirBC
```

[ ]:
```
ke = Constant(2e-4)

f = Constant(0.2)
u = TrialFunction(V)
v = TestFunction(V)

a = ke * inner(grad(u), grad(v)) * dx
L = dot(f, v) * dx

u_sol = Function(V)

solve(a == L, u_sol, bcs)
```

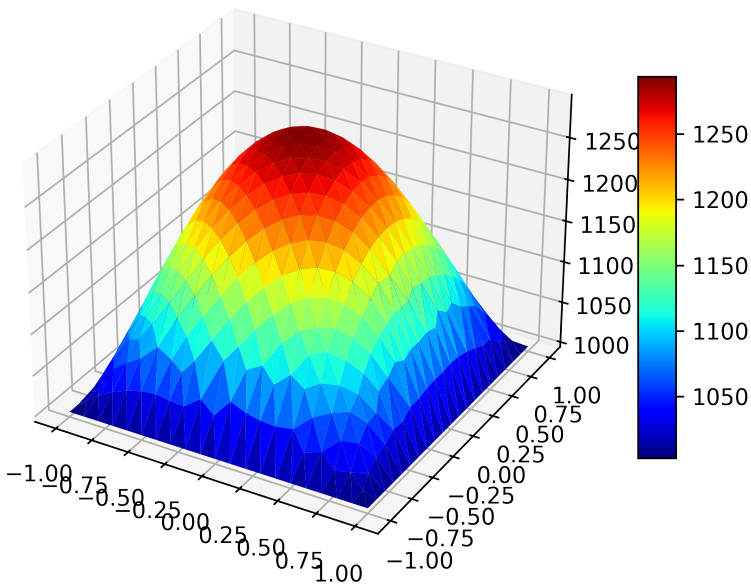
[ ]:
```
coordinates = mesh.coordinates()
values = u_sol.compute_vertex_values(mesh)
x, y = coordinates[:, 1], coordinates[:, 0]

triang = Triangulation(x, y, mesh.cells())
plt.figure(figsize=(6,5))

plt.tricontourf(triang, values, cmap='jet')
plt.colorbar()
plt.triplot(triang, 'k-', lw=0.5)
plt.xlabel('x')
plt.ylabel('y')
plt.savefig(f'heat_2d', dpi=300, transparent=True); plt.show()
```

```
point = Point(0.25, 0.25)
u_value = u_sol(point)
print(f"Solution at point {point}: {u_value}")
```




```
Solution at point <dolfin.cpp.geometry.Point object at 0x7fd76fd9c430>:
1263.3179567734976
```

[ ]:
```
fig = plt.figure(figsize=(5, 5))
ax = fig.add_subplot(111, projection='3d')

x, y, z = triang.x, triang.y, values

ax.plot_trisurf(x, y, z, triangles=triang.triangles, cmap='jet',␣
 ↪edgecolor='none')

mappable = ax.collections[0]  # Extract the mappable object

plt.tight_layout()
fig.colorbar(mappable, ax=ax, shrink=0.5, aspect=10)
plt.savefig('heat_3d', dpi=300, transparent=True)
plt.show()
```



The code can be accessed via Google Colab.

Summary

This coding procedure outlined a systematic approach to simulating and visualizing heat conduction in a square domain using Python. The key steps included:

- Defining the computational domain with clear specifications for the grid and material properties.
- Applying boundary conditions to model the physical constraints accurately.
- Solving the governing equations using numerical methods for heat transfer.
- Visualizing the results to gain insights into the temperature distribution across the domain.

Students are encouraged to experiment with the provided framework by modifying the boundary conditions, such as changing the fixed temperatures or implementing insulated boundaries. Observing the resulting changes in temperature distribution provides a deeper understanding of how boundary conditions influence the system's behavior. This iterative process fosters critical thinking and reinforces concepts of heat transfer and numerical modeling.

Problem Session #8 3/3/2025.

Definition B.1. V: vector space.

a norm is a function: $\|\cdot\|: V \to \mathbb{R}$

such that for $v, u \in V$ & $\alpha \in \mathbb{R}$

1) $\|v\| \geq 0$ & $\|v\| = 0$ IFF $v = 0$

2) $\|\alpha v\| = |\alpha| \, \|v\|$

3) $\|v + u\| \leq \|v\| + \|u\|$

B.5. For $v \in V_1$, H^1-norm

$$\|v\|_{1,2} = \left[\int_a^b v(x)^2 \, dx + \int_a^b v'(x)^2 \, dx \right]^{1/2}$$

$$= \Big[\underbrace{\|v\|_{0,2}^2}_{L^2\text{-norm}} + \underbrace{|v|_{1,2}^2}_{H^1\text{-Seminorm}} \Big]^{1/2}$$

Definition B.2 A vector space V with a norm defined over $\|\cdot\|: V \to \mathbb{R}$ is called a normed space, denoted as $(V, \|\cdot\|)$

B.10. Let $\Omega \subset \mathbb{R}^n$, $n \in \mathbb{N}$. for such domain Ω, the norm $\|v\|_{0,2}$ of $v : \Omega \to \mathbb{R}$ is defined as:

$$\|v\|_{0,2} = \left[\int_\Omega v(x)^2 \, d\Omega \right]^{1/2}$$

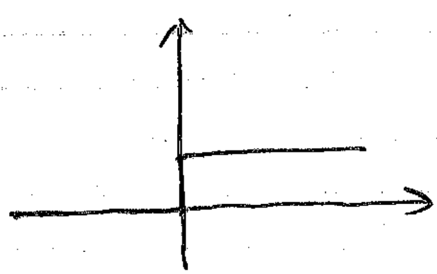
The set

$$L^2(\Omega) = \{ v : \Omega \to \mathbb{R} \mid \|v\|_{0,2} < \infty \}$$

is called the $L^2(\Omega)$ space, and $(L^2(\Omega), \|\cdot\|_{0,2})$ is a normed space. The space $L^2(\Omega)$ is said to contain all square-integrable functions.

→ does not need to be smooth. e.g.,

$\Omega = [-1, 1]$ contains $H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$,



integral of the square of the abs. value is finite.

However, $H(x) \notin L^2(\mathbb{R})$.

↳ why? $\int_{-\infty}^{+\infty} (H(x))^2 \, dx = \int_0^{+\infty} 1 \, dx = +\infty$

B.11 let $\Omega \subset \mathbb{R}^n$, $n \in \mathbb{N}$. For such domain Ω, we define H^1-norm:

$$\|v\|_{1,2} = \left[\|v\|_{0,2}^2 + \sum_{i=1}^{n} \left\| \frac{\partial v}{\partial x_i} \right\|_{0,2}^2 \right]^{1/2}$$

$\rightarrow$ we define $H'(\Omega)$-space

$$H'(\Omega) = \{ v : \Omega \to \mathbb{R} \mid \|v\|_{1,2} < \infty \}.$$

normed space: $(H'(\Omega), \|\cdot\|_{1,2})$

Functions in $H'(\Omega)$ contain "both function & each one of its partial derivatives is square integrable". Alternatively, the function & each of its partial derivatives is in $L^2(\Omega)$.

if a function $v \in H'(\Omega)$, then $v \in L^2(\Omega)$.

e.g., lee $\Omega = [-1, 1] \times [-1, 1]$.

① function $v(x_1, x_2) = x_1^2 + x_2^3 \in H'(\Omega)$,

$$\|v\|_{1,2}^2 = \int_{-1}^{1}\int_{-1}^{1} (x_1^2 + x_2^3)^2 \, dx_1 dx_2 + \int_{-1}^{1}\int_{-1}^{1} (2x_1)^2 dx_1 dx_2$$

$$+\int_{-1}^{1}\int_{-1}^{1}(3x_2^2)^2\,dx_1dx_2 = \frac{292}{21} < \infty$$

② the function $v(x_1, x_2) = \ln(1+x_1) + \ln(1+x_2)$

$\notin H^1(\Omega)$, but $v \in L^2(\Omega)$, since

$$\|v\|_{0,2}^2 = \int_{-1}^{1}\int_{-1}^{1}\left(\ln(1+x_1) + \ln(1+x_2)\right)^2 dx_1dx_2$$

$$= 24 + 8\ln(4)(\ln(2) - 2) < \infty$$

$$\|v\|_{1,2}^2 = \|v\|_{0,2}^2$$

$$+ \int_{-1}^{1}\frac{1}{(1+x_1)^2}\,dx_1dx_2 + \int_{-1}^{1}\frac{1}{(1+x_2)^2}\,dx_1dx_2$$

$$= \infty$$

~ A Simple example

$$-u''(x) = \overset{=x}{f(x)} \quad \text{on } [0, 1] \quad \cdots \text{1D}.$$

$$\begin{cases} u''(x) = x, & x \in (0,1). \\ u(0) = 0, & u(1) = 0 \end{cases}$$

Weak form.

$$\int_0^1 (-u''(x))v(x)\,dx = \int_0^1 x\,v(x)\,dx$$

$$\rightarrow \int_0^1 u'(x)\,v'(x)\,dx = \int_0^1 x\,v(x)\,dx.$$

bilinear form.

$$a(u,v) = \int_0^1 u'(x)v'(x)\,dx.$$

linear functional

$$l(v) = \int_0^1 x\,v(x)\,dx$$

bilinear $a(\cdot,\cdot)$ ...

$\rightarrow$ Continuity: $|a(u,v)| \leq \|u'\|_{L^2(0,1)} \|v'\|_{L^2(0,1)}$.

$\rightarrow$ Coercivity: $\int_0^1 |u'(x)|^2\,dx \geqslant \alpha \|u\|^2_{H_0^1(0,1)}$.

for some $\alpha > 0$, Thus $a(u,v) \geqslant \alpha\|u\|^2$.

giving the strict positivity needed for invertibility.

→ Céa's Lemma.

$$\| u - u_h \|_{H_0^1} \leq \left(1 + \frac{M}{\alpha}\right) \min_{v_h \in V_h} \| u - v_h \|_{H_0^1}.$$

M: continuity constant,

α: coercivity constant.

→ in practice, $\min_{v_h} \| u - v_h \|$ is the "best approximation error" of u by FEM span V_h.

~ Convergence rate

For a Poisson-type problem, with P_k-element, with mesh size h.

→ exact sol'n of u is smooth

→ homogeneous Dirichlet B.C.s

H^1-seminorm: $\| u - u_h \|_{H^1(\Omega)} = O(h^k)$

L^2-norm: $\| u - u_h \|_{L^2(\Omega)} = O(h^{k+1})$.

L^1-norm: $\| u - u_h \|_{L^1(\Omega)} = O(h^{k+1})$ → L^r $1 \leq r \leq 2$

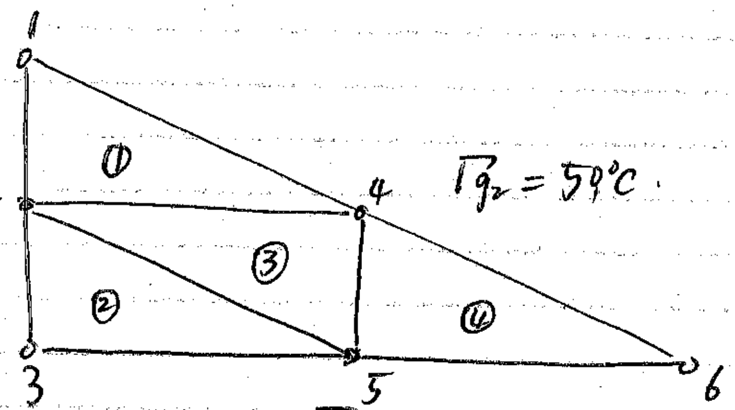
$$\|v\|_{L^1(\Omega)} \le \sqrt{|\Omega|}\, \|v\|_{L^2(\Omega)}.$$

finite measure of the domain.

$$\|u - u_h\|_{L^1(\Omega)} \le \sqrt{|\Omega|}\, \|u - u_h\|_{L^2(\Omega)} = \mathcal{O}(h^{k+1}).$$

Further Questions to explore: what if the assumptions do not hold ... ?

Problem Session #9. (Final Review).



$100°C = \Gamma_{g_1}$

$= \{ (x_1, x_2)$

$\Omega \mid x_1 = 0 \}$

$\Gamma_{g_2} = 50°C$.

$\Gamma_{g_2} = \partial\Omega \setminus (\Gamma_{g_1} \cup \Gamma_h)$

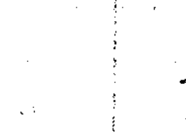
Γ_h.

$\parallel$

$\{ (x_1, x_2) \in \partial\Omega \mid x_2 = 0,\ 0 < x_1 < 6 \}$.

find temperature $T : \Omega \to \mathbb{R}$ s.t.

$$
\begin{aligned}
-\operatorname{div}(K(x)\nabla T) &= 0 && \text{on } \Omega \\
T &= 100°C && \text{on } \Gamma_{g_1} \\
T &= 50°C && \text{on } \Gamma_{g_2} \\
K(x)\nabla T \cdot \check{n} &= 0 && \text{on } \Gamma_h
\end{aligned}
$$



→ Construct variational equation of T:

$$
-\int_\Omega \operatorname{div}(K\nabla T)\, v\, d\Omega = \int_\Omega (K\nabla T)\cdot \nabla v\, d\Omega
$$

$$
- \int_{\Gamma_g} \cancel{(K\nabla T)\cdot \check{n}}^{\,0}\, v\, d\Gamma_h
$$

$$a(T, v) = \int_\Omega (K \nabla T) \cdot \nabla v \, d\Omega \qquad , \forall v \in \mathcal{V}.$$

$$\ell(v) = 0$$

$$\mathcal{V} = \{ v : \Omega \to \mathbb{R} \text{ Smooth} \mid v = 0, \ \forall \underline{x} \in T_{g_1}, T_{g_2} \}.$$

... Define $\mathcal{W}_h$, $\mathcal{V}_h$ & $\mathcal{S}_h$ → state the FEM

$$\mathcal{W}_h = \text{span}(N_1, N_2, N_3, N_4, N_5, N_6).$$

$$\mathcal{V}_h = c N_5 \quad , \quad c \in \mathbb{R}.$$

$$\mathcal{S}_h = T_{g_1}(N_1 + N_2 + N_3) + T_{g_2}(N_4 + N_6) + c N_5.$$

* Find $T_h \in \mathcal{S}_h$ such that

$$a(T_h, v_h) = \ell(v_h) \quad \forall v_h \in \mathcal{V}_h.$$

$$a(T_h, v_h) = \int_\Omega (K \nabla T_h) \cdot \nabla v_h \, d\Omega.$$

$$\ell(v_h) = 0.$$

→ Find LV.

"anti-clockwise"

$$LV = \begin{bmatrix} 1 & 2 & 2 & 4 \\ 2 & 3 & 5 & 5 \\ 4 & 5 & 4 & 6 \end{bmatrix}$$

Assume thermal conductivity is constant for each element ($\underline{\underline{K}}(\underline{x}) \approx k^e \underline{\underline{I}}$, $\underline{x} \in \Omega^e$, $k^e \in \mathbb{R}$).

expressions of N_a^e & A are provided.

k^e	Value
k^1	14
k^2	27
k^3	45
k^4	27

$$N_1^e = \frac{1}{2A}\left[-(X_2^3 - X_2^2)(x_1 - X_1^2) + (X_1^3 - X_1^2)(x_2 - X_2^2)\right]$$

$$N_2^e = \frac{1}{2A}\left[-(X_2^1 - X_2^3)(x_1 - X_1^3) + (X_1^1 - X_1^3)(x_2 - X_2^3)\right]$$

$$N_3^e = \frac{1}{2A}\left[-(X_2^2 - X_2^1)(x_1 - X_1^1) + (X_1^2 - X_1^1)(x_2 - X_2^1)\right]$$

$$A = \frac{1}{2}(X_1^2 - X_1^1)(X_2^3 - X_2^1) - (X_2^2 - X_2^1)(X_1^3 - X_1^1)$$

Constrained index

$\eta_g = \{1, 2, 3, 4, 6\}$. $\quad l(v_h)=0$

We can write out K and F

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad F = \begin{bmatrix} T_{\Gamma g_1} \\ T_{\Gamma g_1} \\ T_{\Gamma g_1} \\ T_{\Gamma g_2} \\ 0 \\ T_{\Gamma g_2} \end{bmatrix}$$

$LV = LG$ (conformal) $\rightarrow K^e_{ab} \rightarrow K_{LG(a,e)\,LG(b,e)}$.

therefore

$$K_{51} = 0$$

$$K_{52} = K^2_{31} + K^3_{21}$$

$$K_{53} = K^2_{32}$$

$$K_{54} = K^3_{23} + K^4_{21}$$

$$K_{55} = K^2_{33} + K^3_{22} + K^4_{22}$$

$$K_{56} = K^4_{23}$$

$A \rightarrow$ same for all elements $\rightarrow$ Assume k const.

$$K^e_{ab} = \int_{\Omega^e} k^e \nabla N^e_b \cdot \nabla N^e_a \, d\Omega^e = k^e \nabla N^e_b \cdot \nabla N^e_a \int_{\Omega^e} d\Omega^e$$

$$= k^e A \nabla N^e_b \cdot \nabla N^e_a$$

Write out the gradients:

$$\nabla N_1^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \nabla N_2^2 = \begin{bmatrix} -1/3 \\ -1 \end{bmatrix} \qquad \nabla N_3^2 = \begin{bmatrix} 1/3 \\ 0 \end{bmatrix}$$

$$\nabla N_1^3 = \begin{bmatrix} -1/3 \\ 0 \end{bmatrix} \qquad \nabla N_2^3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \qquad \nabla N_3^3 = \begin{bmatrix} 1/3 \\ 1 \end{bmatrix}$$

$$\nabla N_1^4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \nabla N_2^4 = \begin{bmatrix} -1/3 \\ -1 \end{bmatrix} \qquad \nabla N_3^4 = \begin{bmatrix} 1/3 \\ 0 \end{bmatrix}$$

Substituting back into K_{ij}

$$K_{51} = 0$$

$$K_{52} = K_{31}^2 + K_{21}^3 = 0$$

$$K_{53} = K_{32}^2 = -3A.$$

$$K_{54} = K_{23}^3 + K_{21}^4 = -72A.$$

$$K_{55} = K_{33}^2 + K_{22}^3 + K_{22}^4 = 78A.$$

$$K_{56} = K_{23}^4 = -3A.$$

$$0 = T_{B_1}(K_{51} + K_{52} + K_{53}) + T_{B_2}(K_{54} + K_{56}) + u_5 K_{55}$$

$$0 = 100(0 + 0 - 3A) + 50(-72A - 3A) + u_5 78A.$$

$$u_5 = 675/13$$

we have:

$$T_h = 100(N_1 + N_2 + N_3) + 50(N_4 + N_6) + \frac{675}{13} N_5$$

$\rightarrow$ From T_h, find values @ centroid of element $\bar{x}^e$.

$\bar{x}^e$
$(1, 4/3)$
$(1, 1/3)$
$(2, 2/3)$
$(4, 1/3)$

$$T_h(\bar{x}^1) = \frac{1}{3}(100 + 100 + 50).$$

$$T_h(\bar{x}^2) = \frac{1}{3}\left(100 + 100 + \frac{675}{13}\right)$$

$$T_h(\bar{x}^3) = \frac{1}{3}\left(100 + 50 + \frac{675}{13}\right)$$

$$T_h(\bar{x}^4) = \frac{1}{3}\left(50 + 50 + \frac{675}{13}\right).$$

$\rightarrow$ What convergence rates r_i would you expect $\|T - T_h\|_{0,2,\Omega}$ & $\|T - T_h\|_{1,2,\Omega}$

$r(\|T - T_h\|_{0,2,\Omega}) = k+1 = 2.$ $\leftarrow$ 1st order elem.

$r(\|T - T_h\|_{0,2,\Omega}) = k = 1$

Now lets switch to P_2 - element.

assume you have access to thermocouple allows you to get measurement T_{meas} @ $\underline{X}_{meas}$.

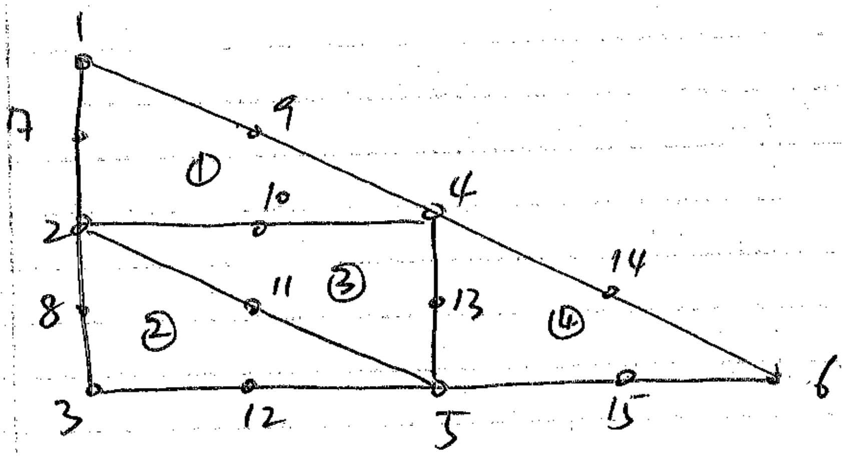
$$\hookrightarrow \quad -\operatorname{div}(k(\underline{x})\nabla T) = 0 \qquad \text{on } \Omega$$

$$T = 100^\circ C \qquad \text{on } \Gamma_{g_1}$$

$$T = 50^\circ C \qquad \text{on } \Gamma_{g_2}.$$

$$k(x)\, \nabla T \cdot \check{n} = 0. \qquad \text{on } \Gamma_h.$$

$$T(\underline{x}) = T_{meas}(\underline{X}_{meas})$$



$$T_h = 100(N_1 + N_2 + N_3 + N_7 + N_8) + 50(N_4 + N_6 + N_9 + N_{14}) + \sum_{j \in \mathcal{J}_h} T^j_{meas} N_j.$$

$$\mathcal{J}_h = \{5, 10, 11, 12, 13, 15\}$$