§1.2 FEM for the model problem with piecewise linear functions — Galerkin method.

- Model problem:

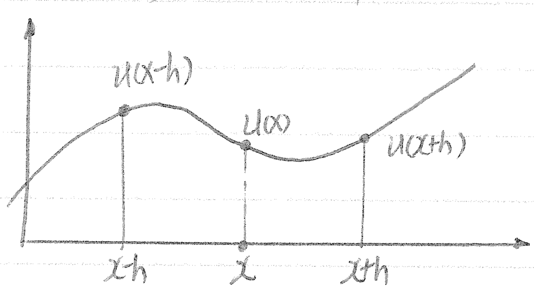
$$\begin{cases} -u''(x) = f(x), & 0<x<1 \\ u(0) = u(1) = 0. \end{cases}$$

continuous.

- Finite Difference Method (FDM):

FDM is based on Taylor series expansion.
FEM is based on projection.

- Approximation derivatives by Finite Difference:



Goal: Approximate $u''(x)$ by $u(x), u(x-h), u(x+h)$:

Taylor expansion:

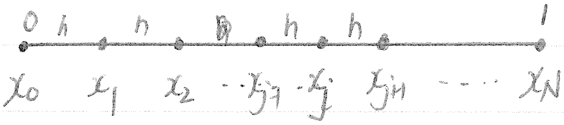
$$\begin{cases} u(x+h) = u(x) + h u'(x) + \frac{h^2}{2} u''(x) + \frac{h^3}{6} u'''(x) + O(h^4) \\ u(x-h) = u(x) - h u'(x) + \frac{h^2}{2} u''(x) - \frac{h^3}{6} u'''(x) + O(h^4) \end{cases}$$

$$\Rightarrow \quad u(x+h) + u(x-h) = 2u(x) + h^2 u''(x) + O(h^4)$$

$$\Rightarrow \quad u''(x) = \frac{u(x+h) - 2u(x) + u(x-h) + O(h^4)}{h^2} = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2)$$

(central difference)

• step 1°: Grid generation, e.g. uniform Cartesian grid.



(Goal: Find approximate values of $u(x)$ at x_j, $j=0,1,\cdots,N$: $U_j \approx u(x_j)$)

• step 2°: FD approximation of derivatives:

At each grid point x_j: $(j=1,2,\cdots,N-1)$

$$-u''(x)\Big|_{x=x_j} = f(x)\Big|_{x=x_j}, \qquad \text{i.e.} \quad -u''(x_j) = f(x_j)$$

$$\Longrightarrow \quad -\frac{u(x_j-h) - 2u(x_j) + u(x_{j+1})}{h^2} \approx f(x_j) \qquad \text{(Not exact; truncation error)}$$

$$\Longrightarrow \quad -\frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} = f_j, \quad j=1,2,\cdots,N-1.$$

• step 3°: Imposition of BCs:

$$u(0) = u(1) = 0 \quad \Longrightarrow \quad U_0 = U_N = 0.$$

$j=1$: $$-\frac{U_0 - 2U_1 + U_2}{h^2} = f_1 \xrightarrow{U_0=0} \frac{2}{h^2}U_1 - \frac{1}{h^2}U_2 = f_1.$$

$1<j<N-1$: $$-\frac{1}{h^2}U_{j-1} + \frac{2}{h^2}U_j - \frac{1}{h^2}U_{j+1} = f_j$$

$j=N-1$: $$-\frac{U_{N-2} - 2U_{N-1} + U_N}{h^2} = f_{N-1} \xrightarrow{U_N=0} -\frac{1}{h^2}U_{N-2} + \frac{2}{h^2}U_{N-1} = f_{N-1}$$

Now, put them together to get a linear system of equation:

$$\begin{pmatrix} \frac{2}{h^2} & -\frac{1}{h^2} & & & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ 0 & & & -\frac{1}{h^2} & \frac{2}{h^2} \end{pmatrix}_{(N-1)\times(N-1)} \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{N-2} \\ U_{N-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{pmatrix}$$

$$AU = F$$

- step 4°: Solve $AU=F$ to get numerical solutions.

- step 5°: Error Analysis; stability analysis; convergency analysis ....

$$|u(x_j) - U_j| \le c h^2.$$

- <u>Remarks: on FDM:</u>

(1) Simple to use and understand

(2) very easy to implement for rectangular regions in 2D/3D.

(3) point-wise

(4) difficult for complicated geometries.

- Galerkin FEM method:

• step 1°: reformulation (derive a variational or weak form):

$$(D)\ \begin{cases} -u''=f, & 0<x<1 \\ u(0)=u(1)=0 \end{cases}$$

$\boxed{\begin{array}{l} V\text{: solution space} \\ V_h\text{: Finite element space} \end{array}}$

$\Downarrow$

infinite dimensional.

(V) Find $u \in V$ s.t. $(u', v') = (f, v)$ for $\forall\, v \in V$.

$\Downarrow$

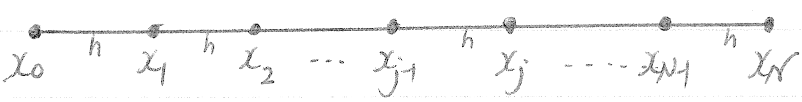
(V_h) Find $u \in V_h$ s.t. $(u', v') = (f, v)$ for $\forall\, v \in V_h$!

Where V_h is a subspace of V (projection!)

Finite dimensional

• step 2°: Triangulation (mesh generation):

discretize the given domain into a collection of preselected finite elements.



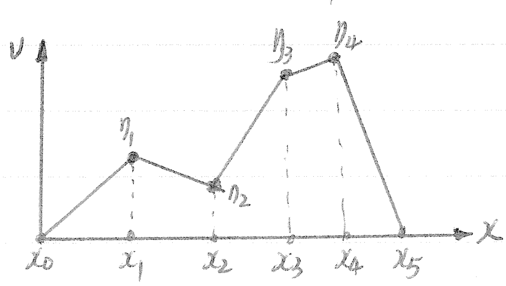
uniform Cartesian grid P

Nodes: x_j, $\cancel{x_1, x_2, \cdots, x_N}$. $j=0,1,\cdots,N$.

Elements: $[x_{j-1}, x_j]$, $j=1,2,\cdots,N$.

• step 3°: construct a finite-dimensional subspace V_h of the space V (choose basis functions of V_h):

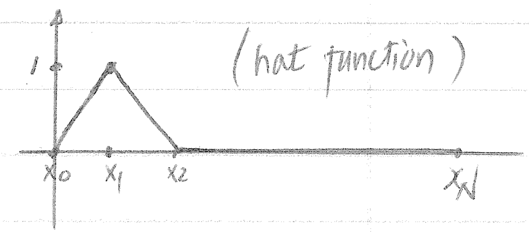
$$V_h = \{ v : v \in V \text{ and } v \text{ is piecewise linear with support } P \}$$



V_h is finite-dimensional: $\dim(V_h) = N-1$!

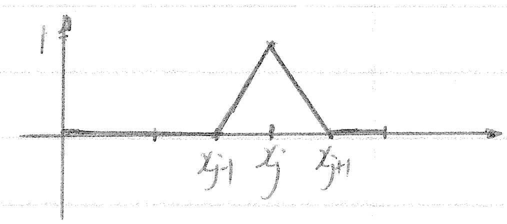
Actually, V_h has basis functions $\phi_1(x), \phi_2(x), \cdots, \phi_{N-1}(x)$!

$$\phi_1(x) = \begin{cases} \frac{x}{h}, & 0 \le x \le x_1 \\ \frac{x_2 - x}{h}, & x_1 \le x \le x_2 \\ 0, & x_2 \le x \le 1 \end{cases}$$



in general:

$$\phi_j(x) = \begin{cases} 0, & x < x_{j-1} \\ \frac{x - x_{j-1}}{h}, & x_{j-1} \le x \le x_j \\ \frac{x_{j+1} - x}{h}, & x_j \le x \le x_{j+1} \\ 0, & x > x_{j+1} \end{cases}$$



Note:
$$\phi_j(x_i) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \ne j, \end{cases} \qquad 1 \le i, j \le N-1$$

Q: Why $\{\phi_1(x), \phi_2(x), \cdots, \phi_{N-1}(x)\}$ is a basis of V_h ?

① Obviously, $\phi_1(x), \phi_2(x), \cdots, \phi_{N-1}(x)$ are linearly independent!

If $c_1\phi_1(x) + c_2\phi_2(x) + \cdots + c_{N-1}\phi_{N-1}(x) = 0$,

then $c_1\phi_1(x_1) + c_2\phi_2(x_1) + \cdots + c_{N-1}\phi_{N-1}(x_1) = 0$.

$\Rightarrow$

$c_1\phi_1(x_1) = c_1 = 0$. similarly, $c_2 = c_3 = \cdots = c_{N-1} = 0$.

② let $v \in V_h$.

Then set $\eta_j = v(x_j)$, $j = 1, 2, \cdots N-1$

Then $v = \eta_1\phi_1(x) + \eta_2\phi_2(x) + \cdots + \eta_{N-1}\phi_{N-1}(x)$!!!

Look for the "best solution" in V_h.

- step 4^0: (~~Derive "element equations" for all typical elements in the mesh~~)

Approximate solution: Let $u_h(x) = \sum_{j=1}^{N-1} c_j \phi_j(x)$ be the solution of

c_j are unknowns!

the weak form (V_h). Namely

$$(u_h', v') = (f, v) \quad \text{for } \forall v \in V_h.$$

Note for this case:
$c_j = u_h(x_j)$
$\approx u(x_j)$
"unknown" is still the approximate values of the $u(x)$ at grids
$\Rightarrow$ Node basis!!!

i.e.

$$\int_0^1 u_h' v' dx = \int_0^1 f v dx \quad \text{for } \forall v \in V_h$$

$$\Longrightarrow \int_0^1 \left(\sum_{j=1}^{N-1} c_j \phi_j(x) \right)' v' dx = \int_0^1 f v dx,$$

$$\Longrightarrow \sum_{j=1}^{N-1} c_j \int_0^1 \phi_j'(x) v'(x) dx = \int_0^1 f v dx, \quad \text{for } \forall v \in V_h.$$

Now, take $v(x) = \phi_1(x), \phi_2(x), \cdots, \phi_{N-1}(x)$ respectively,

$$\begin{cases} \sum_{j=1}^{N-1} c_j (\phi_j', \phi_1') = (f, \phi_1) \\ \sum_{j=1}^{N-1} c_j (\phi_j', \phi_2') = (f, \phi_2) \\ \vdots \\ \sum_{j=1}^{N-1} c_j (\phi_j', \phi_{N-1}') = (f, \phi_{N-1}) \end{cases}$$

Matrix-vector form:

$$\begin{pmatrix} (\phi_1', \phi_1') & (\phi_2', \phi_1'), & \cdots, & (\phi_{N-1}', \phi_1') \\ (\phi_1', \phi_2') & (\phi_2', \phi_2'), & \cdots & (\phi_{N-1}', \phi_2') \\ \vdots & & & \\ (\phi_1', \phi_{N-1}') & (\phi_2', \phi_{N-1}'), & \cdots & (\phi_{N-1}', \phi_{N-1}') \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{pmatrix} = \begin{pmatrix} (f, \phi_1) \\ (f, \phi_2) \\ \vdots \\ (f, \phi_{N-1}) \end{pmatrix}$$

$$AU = F.$$

For hat function,

$$(\phi_j', \phi_j') = \int_{x_{j-1}}^{x_{j+1}} \phi_j' \phi_j' \, dx = \int_{x_{j-1}}^{x_j} \left(\frac{1}{h}\right)^2 dx + \int_{x_j}^{x_{j+1}} \left(-\frac{1}{h}\right)^2 dx$$

$$= \frac{2}{h}.$$

$$(\phi_j', \phi_{j+1}') = \int_{x_j}^{x_{j+1}} \phi_j' \phi_{j+1}' \, dx = \int_{x_j}^{x_{j+1}} \left(-\frac{1}{h}\right)\left(\frac{1}{h}\right) dx = -\frac{1}{h}.$$

$$(\phi_{j+1}', \phi_j') = -\frac{1}{h}$$

$$(\phi_i', \phi_j') = 0 \quad \text{if } |i-j| \geq 2.$$

~~We have~~

Assume: $f(x) \equiv f(x_j)$ in $[x_{j-1}, x_j]$

$$(f, \phi_j) = \int_{x_{j-1}}^{x_j} f(x)\phi_j(x) \, dx \approx f(x_j) \int_{x_{j-1}}^{x_j} \phi_j \, dx = h f(x_j)$$

so

$$\begin{pmatrix} \frac{2}{h} & -\frac{1}{h} & & 0 \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & \\ & \ddots & \ddots & \ddots \\ 0 & & -\frac{1}{h} & \frac{2}{h} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{pmatrix} = h \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \end{pmatrix}$$

(Stiffness matrix) $AU = F$ load vector

is the same as the finite difference method!

- step 5°: solve the system $AU=F$.

- step 6°: post-processing of the results !!!

- Remarks on FEM:

① very successful for mechanical problems
② successful commercial packages
③ Natural approach for complicated boundaries
④ Nice theoretical results at least for elliptic type problems
⑤ "weaker" requirement for the regularities of the solutions
⑥ Need expert knowledge to generate the triangulation:
 (solution: use packages).

- Questions:

(1) Why / how can we change PDE/ODE to a weak form?
(2) How to choose basis function for V_h.
(3) How to programming?
(4) How to solve $AU = F$?

- Properties of the stiffness matrix:

① sparse : ~~[illegible]~~ tri-diagonal
② symmetric
③ positive define: $\Rightarrow$ Non-singular $\Rightarrow$ $AU=F$ has unique sol.
 $(\vec{x}, A\vec{x}) > 0$.

"Iterative method"!
"parallel computing"

Ritz FEM method.

$$(D)\begin{cases} -u''=f, & 0<x<1 \\ u(0)=u(1)=0 \end{cases}$$

$\Downarrow$ Define $(u,v)=\int_0^1 uv\,dx$; Define solution space V.
Define $F(v)=\frac{1}{2}(v',v')-(f,v)$.

(M) Find $u\in V$, such that $F(u)\le F(v)$ for $\forall v\in V$.

!!! Not every problem has a minimization problem !!!

$\Downarrow$ Finite element space: $V_h\subset V$.

(M_h) Find $u\in V_h$, such that $F(u)\le F(v)$ for $\forall v\in V_h$.

• step 1°: re-formulation (derive a minimization problem).

• step 2°: Triangulation (mesh generation)
The same as that in Galerkin method.

• step 3°: Choose a finite-dimensional finite element subspace V_h of the solution space V.
Choose basis functions of V_h.

(see: Galerkin method)

• step 4°: look for the best solution in V_h:

Approximate solution:

Let $u_h(x)=\sum_{j=1}^{N-1} c_j\phi_j(x)$ be the solution of (M_h).

namely,

$$F(u_h) \le F(v), \quad \text{for } \forall\, v \in V_h.$$

Note

$$F(u_h) = \frac{1}{2}(u_h', u_h') - (f, u_h)$$

$$= \frac{1}{2}\int_0^1 (u_h')^2\,dx - \int_0^1 f u_h\,dx$$

$$= \frac{1}{2}\int_0^1 \Big(\sum_j c_j \phi_j'(x)\Big)^2 dx - \int_0^1 f\Big(\sum_j c_j \phi_j(x)\Big)\,dx$$

$$=\!=\!=\!=\; F(c_1, c_2, \cdots, c_{N-1}).$$

i.e.

$F(u_h)$ is a multivariable function of $c_1, c_2, \cdots, c_{N-1}$.
Now, the minimization problem (M_h) is equivalent to this minimization problem:

Find the global minimum of $F(c_1, c_2, \cdots, c_{N-1})$.

$$\Longrightarrow \begin{cases} \dfrac{\partial F}{\partial c_1} = 0 \Rightarrow \displaystyle\int_0^1 \Big(\sum_{j=1}^{N-1} c_j \phi_j'(x)\Big)\phi_1'(x)\,dx - \int_0^1 f\,\phi_1(x)\,dx = 0. \\ \dfrac{\partial F}{\partial c_2} = 0 \Rightarrow \displaystyle\int_0^1 \Big(\sum_{j=1}^{N-1} c_j \phi_j'(x)\Big)\phi_2'(x)\,dx - \int_0^1 f\,\phi_2(x)\,dx = 0. \\ \quad\vdots \\ \dfrac{\partial F}{\partial c_j} = 0 \Rightarrow \displaystyle\int_0^1 \Big(\sum_{j=1}^{N-1} c_j \phi_j'(x)\Big)\phi_j'(x)\,dx - \int_0^1 f\,\phi_j(x)\,dx = 0 \\ \quad\vdots \\ \dfrac{\partial F}{\partial c_{N-1}} = 0 \Rightarrow \displaystyle\int_0^1 \Big(\sum_{j=1}^{N-1} c_j \phi_j'(x)\Big)\phi_{N-1}'(x)\,dx - \int_0^1 f\,\phi_{N-1}(x)\,dx = 0 \end{cases}$$

$$\Longrightarrow \begin{pmatrix} (\phi_1', \phi_1'), & (\phi_2', \phi_1'), & \cdots\cdots, & (\phi_{N-1}', \phi_1') \\ (\phi_1', \phi_2'), & (\phi_2', \phi_2'), & \cdots\cdots, & (\phi_{N-1}', \phi_2') \\ \vdots & \vdots & & \vdots \\ (\phi_1', \phi_{N-1}'), & (\phi_2', \phi_{N-1}') & \cdots & (\phi_{N-1}', \phi_{N-1}') \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{pmatrix} = \begin{pmatrix} (f, \phi_1) \\ (f, \phi_2) \\ \vdots \\ (f, \phi_{N-1}) \end{pmatrix}$$

$$AU = F.$$

! It is exactly the same as the Galerkin method !
using the weak form!

· step 5°: solve $AU=F$ to get $\vec{U}=(c_1, c_2, \cdots, c_N)^T$.

· Remark:

① For self-adjoint elliptic problem, (M) and (V) are equivalent.

② In general.

$$(M) \longrightarrow (V)$$

③ Not all problem have minimization problem.

④ Many problem have weak forms, for example.

PDE: $f(u, u_x, u_y) = 0$

Weak form: $\int_\Omega f(u, u_x, u_y)\, v \, d\Omega = 0$, for $\forall v \in V$.

⑤ We can always use Galerkin form; whether the method will converge is a different story.