The Finite Element Method (FEM)

FEM is a systematic way of getting approximated solutions to PDEs.

Let us first look at a 1D problem (this is covered in §11.5 B&F)

$$(BVP)\quad \begin{cases} -\dfrac{d^2u}{dx^2} = f \\ u(0) = u(1) = 0 \end{cases}$$

← Other B.C. can be considered but we choose homog. Dirichlet B.C. for simplicity.

Let v be a sufficiently smooth test function w/ $v(0) = v(1) = 0$.
Multiplying on both sides by v and integrating we get:

$$(f, v) = \int_0^1 f v\, dx = -\int_0^1 u'' v\, dx \underset{IBP}{=} \underbrace{-u'v\Big|_0^1}_{=0} + \int_0^1 u'v'\, dx$$

We reformulate (BVP) in the variational or weak form:

> Find $u \in V$ s.t.
> $$a(u, v) = (f, v) \quad \forall v \in V.$$ (WF)

Here the space V of functions is:

$$V = \left\{ v \;\middle|\; \int_0^1 v^2\, dx < \infty,\ \int_0^1 (v')\, dx < \infty \text{ and } v(0) = v(1) = 0 \right\}$$

Notice $v \in V$ satisfies B.C. automatically. Also def of V needs to be adjusted to particular B.C.

Here we used the bilinear form or energy inner product:

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

This def. of bilin. form needs to be adjusted to particular problem and could be something more complicated (see 2D).

Question: are weak formulation and BVP the same?

Not really:
- a solution to BVP must have a second derivative (f)
- Solving (WF) allows us to make sense of what is a "solution" to BVP when we only have one derivative.

Although there are solutions to (WF) that are not solutions to (BVP) one can show that:

Theorem: Let $f \in C^0([0,1])$ and $u \in C^2([0,1])$ solve (WF), then u solves (BVP).

So when solution is smooth enough: (BVP) $\Leftrightarrow$ (WF)

Ritz Galerkin approximation: This is the theoretical basis for FEM but can be used in other settings.

Let $S \subset V$ be a finite-dimensional subspace of V: This is the "search space" that we seek the solution in and also the "test function space" that we will replace V in (WF).

The Ritz Galerkin approximation:

Find $u_S \in S$ s.t.

$$a(u_S, v) = (f, v) \quad \forall v \in S \qquad \text{(RG)}$$

Notice how close this is to (WF). The difference is that we now have a finite number of equations to enforce instead of an infinite number of eq. as in (WF).

Let $\{\phi_i\}_{i=1}^n$ be a basis of S. This means that any function $v \in S$ can be written as:

$$v(x) = \sum_{i=1}^{n} v_i \phi_i(x)$$ and this decomposition is unique.

(Yet another way: $S = \text{span}\{\phi_i\}_{i=1}^n$ and $\dim S = n$)

We can solve (RG) by solving a linear system:

Let $u_S = \sum_{i=1}^{n} U_i \phi_i$ ← unknown coeff to be found.

then:

$$a(u_S, v) = \sum_{i=1}^{n} U_i a(\phi_i, v) = (f, v) \quad \forall v \in S$$

↑ by lin. of $a(\cdot,\cdot)$

$$\Leftrightarrow \quad a(u_S, \phi_j) = \sum_{i=1}^{n} U_i a(\phi_i, \phi_j) = (f, \phi_j) \quad \forall j = 1, \dots, n$$

This gives us a linear system:

$$AU = F$$ — Stiffness matrix

where $A \in \mathbb{R}^{n \times n}$ and $A_{ij} = a(\phi_i, \phi_j)$, $i, j = 1, \dots, n$.

$F \in \mathbb{R}^n$ $\quad F_i = (f, \phi_i)$, $i = 1, \dots, n$

$U \in \mathbb{R}^n$ $\quad U_i =$ coeff of u_h in basis $\{\phi_i\}_{i=1}^n$

Question: This is nice, but do we know for sure that this system $AU = F$ is solvable?

Yes and the sol. is unique.

proof: For contradiction assume $\exists\, V \neq 0$ s.t. $AV = 0$

Let $v = \sum_{i=1}^{n} V_i \phi_i$ then

$$a(v, \phi_j) = 0 \qquad j = 1 \dots n$$

$$\Rightarrow \sum_{i=1}^{n} V_i\, a(v, \phi_i) = 0$$

$$\Rightarrow a(v, v) = 0$$

$$\| \int_0^1 (v')^2 \, dx$$

$\Rightarrow v' = 0$ on $[0,1]$ (we use continuity)

but we know $v(0) = 0 \Rightarrow v(x) = 0$ in $[0,1]$

$\Rightarrow V = 0$, a contradiction. QED.

The stiffness matrix A has very nice properties:

- $A = A^T$ because $a(u,v) = a(v,u)$
- A is positive definite:

$$V^T A V = \sum_{ij} V_i V_j A_{ij} = a(v,v) \geq 0$$

where $v = \sum_{i=1}^{n} V_i \phi_i$.

Moreover: $V^T A V = 0 \Rightarrow v = 0 \Rightarrow V = 0$

↑ using same argument as before.

Remark: One big advantage of FEM over Finite Differences is that the solution we get is a function and not just a collection of points. We can evaluate this function at any point (not only the grid points as in finite diff.) and we can differentiate it, integrate it etc...

Moreover there is a well established machinery to get error estimates for the function & its derivatives.

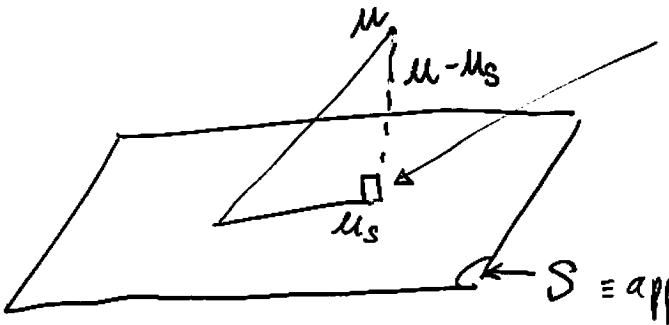
The Basic or fundamental error estimate:

Galerkin orthogonality:

$$a(u, w) = (f, w) \quad \forall w \in S \quad (u \text{ solves WF})$$
$$a(u_S, w) = (f, w) \quad \forall w \in S \quad (u_S \text{ solves RG})$$

$$\Rightarrow \boxed{a(u - u_S, w) = 0 \quad \forall w \in S} \equiv \text{Galerkin } \perp$$

Remember picture (by the way this is same picture as for least square...)



Right angle (orthogonality) is measured with $a(\cdot,\cdot)$ inner prod.

→ Galerkin $\perp$ means error $u - u_S$ is $\perp$ (in the sense of $a(\cdot,\cdot)$) to the approx. space

Now let us introduce the so called energy norm:

$$\|u\|_E = (a(u,u))^{\frac{1}{2}}$$ which satisfies the

Cauchy - Shwarz inequality:

$$|a(v,w)| \leq \|v\|_E \|w\|_E \quad \text{for } v, w \in V.$$

Then for all $v \in S$:

$$\|u-u_S\|_E^2 = a(u-u_S, u-u_S)$$

$$= a(u-u_S, u-v) + \underbrace{a(u-u_S, v-u_S)}_{=0 \text{ by Galerkin} \perp}$$

$$\leq \|u-u_S\|_E \, \|u-v\|_E$$

If $\|u-u_S\|_E \neq 0$ then:

$$\|u-u_S\|_E \leq \|u-v\|_E \quad \forall v \in S$$

$$\Rightarrow \quad \|u-u_S\|_E \leq \inf_{v \in S} \|u-v\|_E$$

But since $u_S \in S$

$$\inf_{v \in S} \|u-v\|_E \leq \|u-u_S\|_E$$

$$\Rightarrow \quad \|u-u_S\|_E = \inf_{v \in S} \|u-v\|_E$$

Since we have an element (namely u_S) in S realizing the infimum we may write:

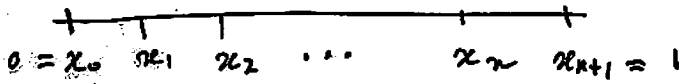
Theorem: $$\|u-u_S\|_E = \min_{v \in S} \|u-v\|_E$$

Translation: RG approx gives best approx. to u in S, if we use as measure the $\|\cdot\|_E$ norm

Before we see what happens with other norms (say L^2 norm) we need to look at a particular example of FEM:

1D P1 elements

Partition interval $[0,1]$



Let S be the linear space of functions v s.t.:

i) $v \in C^0[0,1]$

ii) $v|_{[x_{i-1}, x_i]}$ is a linear poly, for $i = 1, .., n+1$.

iii) $v(0) = v(1) = 0$.

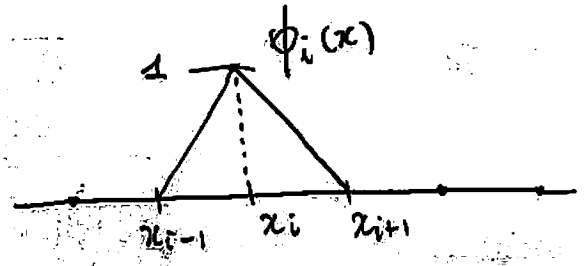
Recall $V = \{ v \mid \int_0^1 v^2 dx < \infty, \int_0^1 (v')^2 dx < \infty \text{ and } v(0) = v(1) = 0 \}$

it is not hard to check that:

$S \subset V$ (v' is defined almost everywhere for a pcws linear fun.)

For convenience we define the basis functions ϕ_i, $i = 1, .., n$ with requirement:

$$\phi_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} = \text{"Kronecker delta"}$$



Note:

$\{\phi_i\}_{i=1}^{n}$ is a basis for S

$\{x_i\}_{i=1}^{n}$ are nodes of discretization.

$\{v(x_i)\}_{i=1}^{n}$ = nodal values of a function v

Any function in S can be written uniquely as:

$$v(x) = \sum_{i=1}^{n} V_i \phi_i(x)$$

and the coefficients in expansion come right away from the δ_{ij} property of the $\{\phi_j\}$:

$$v(x_j) = \sum_{i=1}^{n} V_i \underbrace{\phi_i(x_j)}_{=\delta_{ij}} = V_j$$

$$\Rightarrow \boxed{v(x) = \sum_{j=1}^{n} v(x_j)\,\phi_j(x)}$$

⚠ $v(x)$ is a function, so we can find its value anywhere on $[0,1]$.

For a function v not necessarily in S we define its interpolant.

$$\boxed{v_I(x) = \sum_{i=1}^{n} v(x_i)\,\phi_i(x)}$$

Notice that in general $v \neq v_I$. Equality holds iff $v \in S$.

How well can we approximate a function w/ its interpolant?

Theorem: Let $h = \max_{i=1,\cdots,n+1} x_i - x_{i-1}$. The for all $u \in V$:

$$\boxed{\|u - u_I\|_E \leq C h \|u''\|}$$

Here C is a constant independent of h and u.

(It is of common usage in the FEM litterature to have C a generic positive constant that may be different from one expression to the next)

proof: The proof of this approximation result consists in:

i) Showing <u>local approx</u> result in an element (interval) $[x_{i-1}, x_i]$ (using polynomial approx theory we know from MATH 5610)

ii) Summing up approx errors over all elements.

[**NOTE:** this proof was not seen in class and I am including it just for your general numerical analysis culture]

The step in gives the pcws estimate:

$$(*) \quad \int_{x_{i-1}}^{x_i} [(u-u_I)']^2 dx \leq C (x_i - x_{i-1})^2 \int_{x_{i-1}}^{x_i} [u''(x)]^2 dx$$

Then summing up over elements:

$$\|u-u_I\|_E^2 = \int_0^1 [(u-u_I)']^2 dx = \sum_{i=1}^{n+1} \int_{x_{i-1}}^{x_i} [(u-u_I)']^2 dx$$

$$\leq C \sum_{i=1}^{n+1} (x_i - x_{i-1})^2 \int_{x_{i-1}}^{x_i} [u''(x)]^2 dx$$

$$\leq C h^2 \sum_{i=1}^{n+1} \int_{x_{i-1}}^{x_i} [u''(x)]^2 dx$$

$$= C h^2 \|u''\|^2.$$

$\rightsquigarrow$ result $\|u-u_I\|_E \leq C h \|u''\|$ by taking square root.

To show the pcws estimate $(*)$, we can rewrite it using $e = u - u_I$:

$$\int_{x_{i-1}}^{x_i} [e'(x)]^2 dx \leq C (x_i - x_{i-1})^2 \int_{x_{i-1}}^{x_i} [e''(x)]^2 dx$$

since $u'' = e''$ (as $u_I'' = 0$)

Then we change variables from $[x_{i-1}, x_i]$ to $[0,1]$:

$$x(\hat{x}) = x_{i-1} + \hat{x}(x_i - x_{i-1}), \quad \hat{x} \in [0,1].$$

$$\hat{e}(\hat{x}) = e(x(\hat{x})) = e(x_{i-1} + \hat{x}(x_i - x_{i-1}))$$

c.o.v in integral: $dx = (x_i - x_{i-1})\,d\hat{x}$

$$(x_i - x_{i-1}) \int_0^1 [e'(x(\hat{x}))]^2 d\hat{x} \le C\,(x_i - x_{i-1}) \int_0^1 [e''(x(\hat{x}))]^2\, d\hat{x}$$

Chain rule:

$$\hat{e}'(\hat{x}) = (x_i - x_{i-1})\, e'(x(\hat{x}))$$
$$\hat{e}''(\hat{x}) = (x_i - x_{i-1})^2 e''(x(\hat{x}))$$

Thus:

$$\frac{1}{(x_i - x_{i-1})^2} \int_0^1 (\hat{e}'(\hat{x}))^2\, d\hat{x} \le C\, \frac{1}{(x_i - x_{i-1})^4} (x_i - x_{i-1})^2 \int_0^1 (\hat{e}''(\hat{x}))^2 d\hat{x}$$

So we need to check estimate:

$$\int_0^1 (\hat{e}'(\hat{x}))^2 d\hat{x} \le C \int_0^1 (\hat{e}''(\hat{x}))^2 d\hat{x}$$

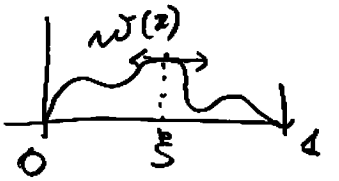
For clarity let us show: $(w = \hat{e})$

$$\int_0^1 (w'(x))^2\, dx \le C \int_0^1 (w''(x))^2 dx$$

Since $w(0) = w(1) = 0$ (by interp prop, error is zero at end points)

we can use Rolle's theorem (or MVT):

$$\exists\ \xi \in (0,1) \text{ s.t. } w'(\xi) = 0$$



$$\Rightarrow w'(y) - \underbrace{w'(\xi)}_{=0} = \int_\xi^y w''(x)\, dx$$

By Cauchy-Schwarz:

$$|w'(y)| = \left| \int_\xi^y w''(x)\,dx \right| = \left| \int_\xi^y w''(x) \times 1 \, dx \right|$$

$$\leq \underbrace{\left| \int_\xi^y 1 \, dx \right|^{\frac{1}{2}}}_{= |y-\xi|^{\frac{1}{2}}} \underbrace{\left| \int_\xi^y [w''(x)]^2 \, dx \right]^{\frac{1}{2}}}_{\leq \left(\int_0^1 [w''(x)]^2 dx \right)^{\frac{1}{2}}}$$

squaring and integrating we get.

$$\int_0^1 |w'(y)|^2 \, dy \leq \underbrace{\int_0^1 |y-\xi| \, dy}_{\leq \frac{1}{2} \text{ for } \xi \in [0,1]} \times \int_0^1 [w''(x)]^2 dx$$

$\Rightarrow$ we get result. QED.

Computer implementation of FEM

Key idea: Compute $A = [a(\phi_i, \phi_j)]_{i,j=1}^{n}$ by summing contributions of each element.

naive FEM implementation

```
for i = 1 .. n
    for j = 1 .. n
        A_ij = a(φ_i, φ_j)
```

Element by element.

```
for e = 1, .. # elements
    A = A + A^e
```

A^e: contribution from element e to stiffness matrix.

$$a(u,v) = \int_0^1 u'v'\,dx = \sum_{e=1}^{n+1} \int_{x_{e-1}}^{x_e} u'v'\,dx$$

$$=: a_e(u,v) = \text{local bilinear form}$$

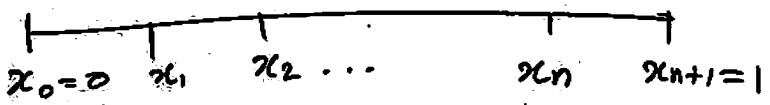
Local bilinear form is easy to evaluate because

$$\phi_i|_{K_e} = 0 \text{ on } K_e \Rightarrow a_e(\phi_i, \phi_j) = 0$$

Moreover $\phi_i|_{K_e} \equiv$ polynomial and integrals w/ polynomials are easy to compute.

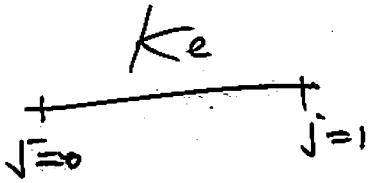
However we need a <u>local to global</u> table map which relates the local degree of freedom from an element to its corresp. global d.o.f.

In our 1D example



Elements $K_e = [x_{e-1}, x_e]$, $e = 1, \ldots, n+1$.

Each element has 2 degrees of freedom (the endpoints).



In order to get the global index associated with node j we can construct the table (matrix):

e ↓ \ j →	0	1
1	1	2
2	2	3
⋮	⋮	⋮
n	n	n+1

or more succintly: $\boxed{i(e,j) = e + j - 1}$

Of course: local to global map depends on particular partition of comp domain.

Now rewrite interpolant of a function f as follows:

$$f_I = \sum_{i=0}^{n+1} f(x_i)\,\phi_i = \sum_{e} \sum_{j=0}^{1} f(x_{i(e,j)})\,\phi_j^e$$

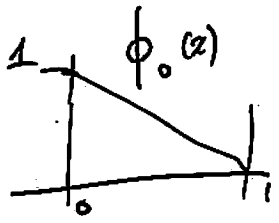
note: this expression is wrong at the nodes, however these mistakes at a few points do not affect result because we integrate.

Here we use notation $\{\phi_j^e\}_{j=0}^1$ for the linear (P_1) basis functions on interval $K_e = [x_{e-1}, x_e]$.

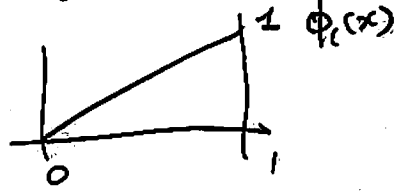
$$\phi_j^e(x) = \phi_j\left(\frac{x - x_{e-1}}{x_e - x_{e-1}}\right)$$

where:

$$\phi_0(x) = \begin{cases} 1-x & x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$



$$\phi_1(x) = \begin{cases} x & x \in [0,1] \\ 0 & \text{otherwise:} \end{cases}$$



So we have related the "local" basis functions ϕ_j^e to those at a reference (parent) element $[0,1]$ with an affine mapping:

$$[0,1] \to [x_{e-1}, x_e]$$

$$x(\hat{x}) = x_{e-1} + \hat{x}(x_e - x_{e-1})$$

$$dx = (x_e - x_{e-1})\,d\hat{x}$$

$$|K_e|$$

Thus:

$$a(u,v) = \sum_e a_e(u,v)$$

where

$$a_e(u,v) = \int_{K_e} u'v'\,dx$$

$$= \int_{K_e} \left(\sum_{j=0}^{1} u_{i(e,j)}\,\phi_j^e\right)' \left(\sum_{j=0}^{1} v_{i(e,j)}\,\phi_j^e\right)' dx$$

Chain rule

$$= \frac{1}{|K_e|^2} \int_{K_e} \left(\sum_{j=0}^{1} u_{i(e,j)}\,\phi_j'\left(\frac{x - x_{e-1}}{x_e - x_{e-1}}\right)\right) \times \left(\sum_{j=0}^{1} v_{i(e,j)}\,\phi_j'\left(\frac{x - x_{e-1}}{x_e - x_{e-1}}\right)\right) dx$$

C.o.V

$$= \frac{1}{|K_e|} \int_0^1 \left(\sum_{j=0}^{1} u_{i(e,j)}\,\phi_j'\right)\left(\sum_{j=0}^{1} v_{i(e,j)}\,\phi_j'\right) d\hat{x}$$

$$\Rightarrow \quad a_e(u,v) = \frac{1}{|k_e|} \begin{bmatrix} v_{i(e,0)} & v_{i(e,1)} \end{bmatrix} A^e_{loc} \begin{bmatrix} u_{i(e,0)} \\ u_{i(e,1)} \end{bmatrix}$$

where $\left(A^e_{loc}\right)_{ij} = \int_0^1 \phi'_{i-1}\, \phi'_{j-1}\, dx, \quad i,j = 1,2.$

$=$ "local" stiffness matrix.

Typical FEM code looks like this:

A = all zero # d.o.f. × # d.o.f. matrix (sparse)

F = all zero # d.o.f. × 1

for $e = 1, \ldots$ # elements

$$A(i(e,:), i(e,:)) = A(i(e,:), i(e,:)) + A^{k_e}_{loc};$$
$$F(i(e,:)) = F(i(e,:)) + F^{k_e}_{loc};$$

where:

$$\left(A^{k_e}_{loc}\right)_{ij} = a_{k_e}\left(\phi^e_i, \phi^e_j\right), \quad i,j = 1, \ldots \text{\# local d.o.f.}$$

$$\left(F^{k_e}_{loc}\right)_i = \left(f, \phi^e_i\right), \quad i = 1, \ldots, \text{\# local d.o.f.}$$

Error estimates revisited

We showed earlier the basic error estimate:

$$\boxed{\|u - u_S\|_E = \min_{v \in S} \|u - v\|_E} \qquad (BE)$$

where $\|u\|_E = (a(u,u))^{\frac{1}{2}}$

and u solves (WF)

u_S solve (RG).

However bound on derivatives may not be practical!

How to get an estimate of the L^2 error:

$$\|u - u_S\|_{L^2} := \left(\int_0^1 (u - u_S)^2 \, dx \right)^{\frac{1}{2}} \quad ?$$

First using (BE) we obtain:

$$\boxed{\|u - u_S\|_E \leq \|u - u_I\|_E \leq C h \|u''\|}$$

$\|u - u_S\|_E \leq \|u - u_I\|_E$ since $u_I \in S$

$\|u - u_I\|_E \leq C h \|u''\|$ where $h = \max_{i=1,\dots,n+1} |x_i - x_{i-1}|$

To get L^2 error estimate we use the so called <u>duality</u> or <u>Aubin-Nitsche</u> trick:

Let w be a sol to:

$$\begin{cases} -w'' = u - u_S & \text{on } [0,1] \\ w(0) = w(1) = 0 \end{cases}$$

Then:

$$\|u-u_S\|^2 = (u-u_S, u-u_S)$$

$$= (u-u_S, w'') \qquad = 0 \text{ by B.C.}$$

$$\overset{\text{IBP}}{=} a(u-u_S, w) - \cancel{(u-u_S)w'\Big|_0^1}$$

$$= a(u-u_S, w-v) \quad \forall v \in S$$

Galerkin $\perp$

Thus using Schwarz inequality:

$$\|u-u_S\|^2 \leq \|u-u_S\|_E \, \|w-v\|_E$$

$$\Rightarrow \quad \|u-u_S\| \leq \|u-u_S\|_E \, \frac{\|w-v\|_E}{\|w''\|}$$

Taking inf. over $v \in S$:

$$\|u-u_S\| \leq \|u-u_S\|_E \inf_{v \in S} \frac{\|w-v\|_E}{\|w''\|}$$

Using interp result:

$$\inf_{v \in S} \|w-v\|_E \leq \|w-w_I\|_E \leq Ch \, \|w''\|$$

$$\Rightarrow \boxed{\|u-u_S\| \leq Ch \|u-u_S\|_E}$$

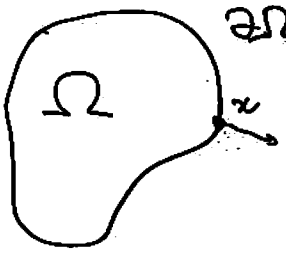
Using interp result again:

$$\boxed{\|u-u_S\| \leq Ch^2 \, \|u''\|}$$

FEM in 2D

Consider the problem:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{cases}$$



$\partial\Omega$ = boundary of Ω

$n(x)$ = outward unit normal to $\partial\Omega$ at x

To find (WF) multiply PDE on both sides by some test function

$$v \in V = \{ v \mid \int_\Omega v^2 dx < \infty, \int_\Omega |\nabla v|^2 dx < \infty, v|_{\partial\Omega} = 0 \}$$

and integrate:

$$-\int_\Omega v \Delta u \, dx = \int_\Omega f v \, dx$$

Then use analogous of IBP, which is Green's identity:

$$\int_\Omega -v \Delta u \, dx = \int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} v (\nabla u \cdot n) dS$$

$= 0$ since $v|_{\partial\Omega} = 0$.

Then we get WF:

> Find $u \in V$ s.t.
>
> $$a(u, v) = (f, v) \quad \forall v \in V$$

(WF)

here: $a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx$

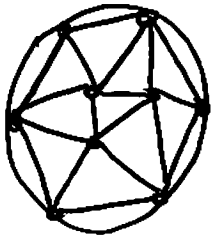
$$(f, v) = \int_\Omega f v \, dx$$

(WF) needs to be adjusted if B.C. are different.

Implementation of 2D P1 elements

First we need to partition Ω into elements, usually triangles or quadrilatera. Here we focus on triangles.

Triangulation



Given coordinates of points in 2D (and even n-dim) it is possible to find a triangulation that maximizes the minimal angle of triangles.

$\Rightarrow$ Delaunay triangulation.

Triangulation is exact i.e. $\bigcup_e K_e = \Omega$, when Ω = cvx polygon.

if it is not, then we are introducing an error.

There is a well-known open source library for computing triangulations called QHULL, which is the engine behind Matlab's delaunay function:

```
tri = delaunay(x,y);   x = vec of x-coord.
                       y = ——— y ———
```

tri =

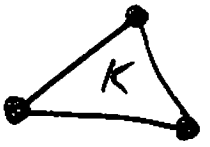
#tri.

tri(e,:) = index of vertices of triangle.

Since error estimates for 2D FEM degrade when triangles are close to degenerate (when they have a small angle) it is possible to adjust position of node (or add / remove) to improve min angle. → many mesh generators

- <u>triangle</u> - J. R. Shewchuck (industrial strength)
- <u>distmesh</u> - Strang and Persson (simple to use and understand)

Since here we are using P1 elements d.o.f are values at vertices.



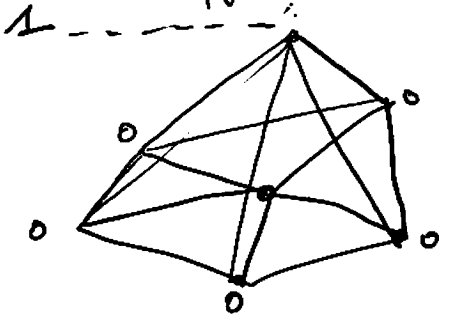
So the local-to-global map is given automatically by the tri array from Delaunay triangulation:

$$tri(e,j) = \text{global idx of } j\text{-th vertex of triangle } e.$$

<u>Basis functions of S:</u>

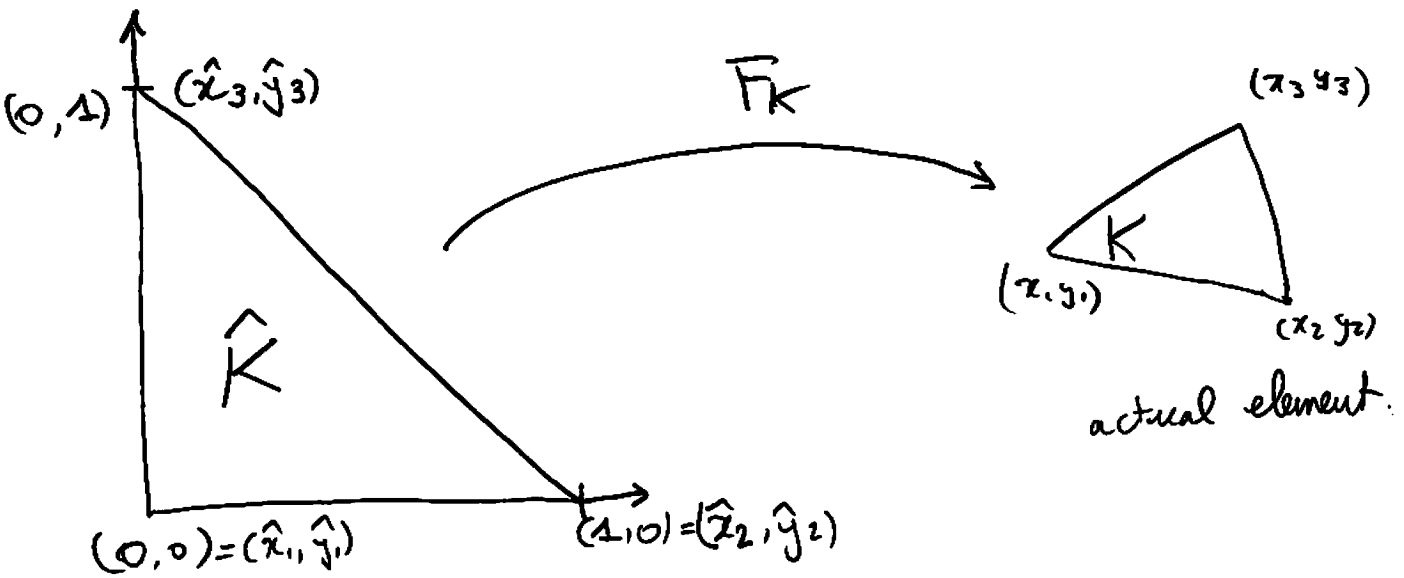
$\phi_j(x)$ = pcws linear on triangulation and s.t.

$$\phi_j(x_i) = \delta_{ij}.$$



→ pyramid shape.

P1 elements are an <u>affine family of elements</u>:



parent ref- } element

$$F_K(\hat{x}, \hat{y}) = B\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} + b = \text{invertible affine mapping}$$

$$= \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Easy to check that:

$$F_K(\hat{x}_i, \hat{y}_i) = (x_i, y_i) \quad i = 1, 2, 3.$$

Also basis functions restricted to element K can be expressed in terms of basis functions at <u>parent element</u> by means of F_K map:

$$\hat{\phi}_i(\hat{x}_j, \hat{y}_j) = \delta_{ij}$$

$$\Rightarrow \quad \begin{cases} \hat{\phi}_1(\hat{x}, \hat{y}) = 1 - \hat{x} - \hat{y} \\ \hat{\phi}_2(\hat{x}, \hat{y}) = \hat{x} \\ \hat{\phi}_3(\hat{x}, \hat{y}) = \hat{y} \end{cases}$$

Local basis function of element K is then:

$$\boxed{\phi_i^K(x,y) = \hat{\phi}_i\left(F_K^{-1}(x,y)\right) = \left(\hat{\phi}_i \circ F_K^{-1}\right)(x,y)}$$

Stiffness matrix

As in 1D we assemble the (global) stiffness matrix by adding the contribution of each element:

for $e = 1, \ldots, \#$ of triangles

$$A(tri(e,:), tri(e,:)) = A(tri(e,:), tri(e,:)) + A^{ke}_{loc\ i}$$

Where local stiffness matrix is:

$$\left(A^{ke}_{loc}\right)_{ij} = a_{ke}\left(\phi_i^{ke}, \phi_j^{ke}\right) = \int_{ke} \nabla\phi_i^{ke} \cdot \nabla\phi_j^{ke}\, dx$$

for $i, j = 1, 2, 3$.

The above loop computes the right thing because:

$$A_{ij} = a(\phi_i, \phi_j) = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j\, dx = \sum_e \int_{ke} \nabla\phi_i|_{ke} \cdot \nabla\phi_j|_{ke}\, dx$$

$$= \sum_{k\,\text{element}} a_K\left(\phi_i^{ke}, \phi_j^{ke}\right)$$

So that if $v, w \in V_h$:

$$a_K(v,w) = a_K\left(\sum_{i=1}^{3} v_{tri(K,i)}\,\phi_i^K, \sum_{j=1}^{3} u_{tri(K,j)}\,\phi_j^K\right)$$

$$= \sum_{i,j=1}^{3} v_{tri(K,i)}\, u_{tri(K,j)}\, a_K(\phi_i^K, \phi_j^K)$$

$$= v_{tri(K,:)}^T\, A_{loc}^K\, u_{tri(K,:)}$$

Local stiffness matrix calculation

Idea: change of variables and chain rule

Recall:

- Change of variables: (c.o.v.)

$$\int_\Omega f(x)\,dx = \int_{\hat\Omega = F(\Omega)} f(F^{-1}(\hat x))\,|\det DF^{-1}|\,d\hat x$$

(check with dilation)

- chain rule (for gradients)

$$\nabla(f\circ F)(x) = DF^T[x]\,(\nabla f)\cdot F(x).$$

Thus:

$$(A_{loc}^K)_{ij} = \int_K \nabla\phi_i^K(x,y)\cdot\nabla\phi_j^K(x,y)\,dxdy$$

$$= \int_K \nabla(\hat\phi_i\circ F_K^{-1})(x,y)\cdot\nabla(\hat\phi_j\circ F_K^{-1})(x,y)\,dxdy$$

Chain rule $=$

$$\int_K \left[DF_K^{-T}\nabla\hat\phi_i(F_K^{-1}(x,y))\right]^T DF_K^{-T}\nabla\hat\phi_j(\cdots)\,dxdy$$

c.o.v. $=$

$$\int_{\hat K} \nabla\hat\phi_i(\hat x,\hat y)^T\left(DF_K^{-1}DF_K^{-T}\right)\nabla\hat\phi_j(\hat x,\hat y)\,|\det DF_K|\,d\hat x\,d\hat y$$

Now the $\hat{\phi}_i \in P_1$, so they have constant gradient!

$$\Rightarrow \text{let } G = \left[D\hat{\phi}_1 \; D\hat{\phi}_2 \; D\hat{\phi}_3 \right] = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Notice:

$DF_K = B_K =$ some matrix = constant over each element.

$$\boxed{A_{loc}^K = \int_{\hat{K}} G^T (B_K^T B_K)^{-1} G \, |\det(B_K)| \, d\hat{x} d\hat{y}}$$

$$= \underbrace{|\hat{K}|}_{=\frac{1}{2}} \, |\det(B_K)| \, G^T (B_K^T B_K)^{-1} G$$

The right hand side can be assembled in a similar way.

Assume for simplicity that $f \in S$. Other more general RHS can be considered by doing more elaborate numerical integration.

Then adding contributions of each element:

for $e = 1 \ldots$ # triangles:

$$F(tri(e,:)) = F(tri(e,:)) + F_{loc}^K,$$

where

$$(F_{loc})_j = \int_K f(x,y) \, \phi_j^K(x,y) \, dx \, dy$$

$$= \int_K \left(\sum_{i=1}^{3} f(x_{tri(K,i)}, y_{tri(K,i)}) \, \phi_i^K(x,y) \right) \phi_j^K(x,y) \, dx dy$$

Then:

$$F_{loc} = M_{loc}^{k} \left[f(x_{tri}(k,:), y_{tri}(k,:)) \right]$$

where $(M_{loc}^{k})_{ij} = \int_{k} \phi_i^k(x,y)\ \phi_j^k(x,y)\, dx\, dy$

These integrals can be evaluated easily on the ref element by C.O.V:

$$(M_{loc}^{k})_{ij} = \int_{\hat{k}} \hat{\phi}_i(\hat{x},\hat{y})\ \hat{\phi}_j(\hat{x},\hat{y})\, |\det DF_k|\, d\hat{x}\, d\hat{y}$$

$$= \frac{|\det DF_k|}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$