M692 F E Sem 1/18:

Ex: Poisson equation in a 2D region

$= \nabla \cdot \nabla u = u_{xx} + u_{yy}$ in x, y

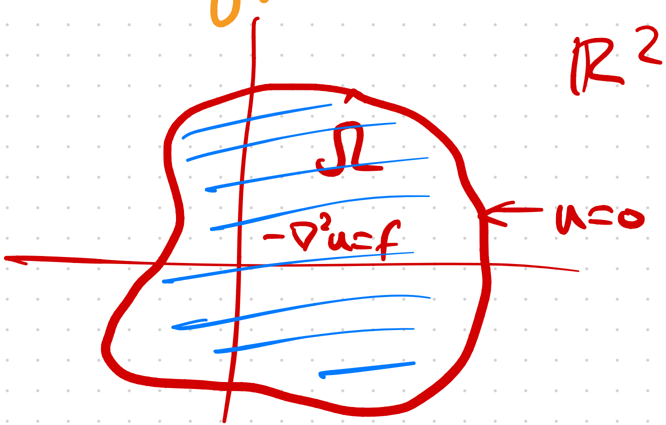
$f(x,y)$

strong form

$-(\nabla^2 u)(x,y) = f(x,y)$ on $\Omega \subset \mathbb{R}^2$

$u(x,y)$

$u = 0$ on $\partial\Omega$ — boundary of Ω



- membrane $u(x,y)$ steady deform
  $f(x,y)$ load
- E&M: $u(x,y)$ = electrostatic potential
  $\vec{E} = \nabla u$ field
  $f(x,y)$ = charge density
- thermo $u(x,y) = T(x,y)$
  steady temperature
  $f(x,y)$ = given heat source
- & diffusion ... at steady state

Weak Form:

$$\boxed{-\nabla^2 u = f} \; \& \; \underset{\text{on } \partial\Omega}{u=0} \Big\} \text{ strong}$$

$$(-\nabla^2 u)\, v = f\, v \qquad \Big\} \text{ mult. by \underline{any} fnctn}$$

$$\iint_\Omega (-\nabla^2 u)(x,y)\, v(x,y)\, dx\, dy = \iint_\Omega f(x,y)\, v(x,y)\, dx\, dy$$

simpler notation

$$\int_\Omega (-\nabla^2 u)\, v\, dx = \int_\Omega f\, v\, dx$$

understandable as inner product in a space of functions prod. rule

$$\nabla\cdot(gX) = \nabla g\cdot X + g\,\nabla\cdot X$$

div. thm (in 2D)

$$\int_\Omega \nabla\cdot X\,dx = \int_{\partial\Omega} X\cdot ds$$

$$\boxed{\int_\Omega (\nabla^2 u)\,v\,dx} = \int_\Omega \nabla\cdot(\nabla u)\,v\,dx$$

$$= \int_\Omega \nabla\cdot(v\nabla u)\,dx - \int_\Omega \nabla v\cdot\nabla u\,dx$$

$$= \boxed{\int_{\partial\Omega} (v\,\nabla u)\cdot ds - \int_\Omega \nabla u\cdot\nabla v\,dx}$$

now assume

$$u, v \in H_0^1(\Omega) = \{ w(x,y) \mid (\nabla w)(x,y) \text{ makes sense, } \& \ w|_{\partial\Omega} = 0 \}$$

So!

prev. slide

because $v = 0$ on $\partial\Omega$

$$\int_\Omega (-\nabla^2 u) v \, dx = -\underbrace{\cancel{\int_{\partial\Omega} v \nabla u \cdot ds}}_{=0} + \int_\Omega \nabla u \cdot \nabla v \, dx$$

$$= \int_\Omega \nabla u \cdot \nabla v \, dx$$

So now weak form:

$$\boxed{\int_\Omega \nabla u \cdot \nabla v \, dv = \int_\Omega f v \, dx \quad \text{for all } v \in H_0^1(\Omega)}$$

Weak form:

$$\int_\Omega \nabla u \cdot \nabla v \, dx - \int_\Omega f v \, dx = 0$$

$\Longleftrightarrow$

Firedrake:

```
from firedrake import *
mesh = UnitSquareMesh(10, 10)
H = FunctionSpace(mesh, 'CG', 1)
u = Function(H)
v = TestFunction(H)    f=Function(H).interpolat
                                      (...)
F = (dot(grad(u), grad(v))
        - f*v)*dx
bcs = ...
solve(F == 0, bcs=bcs)
```