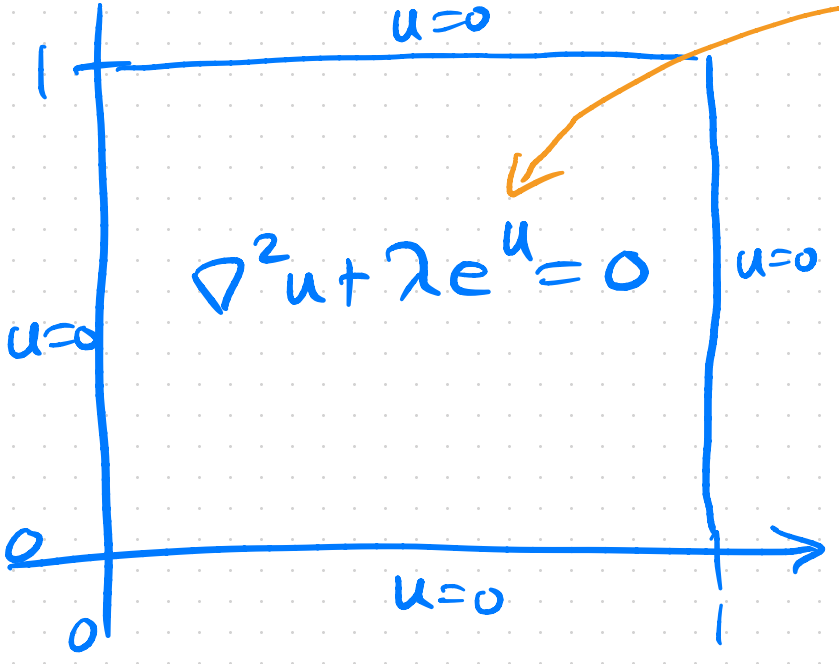
a nonlinear version of Poisson

21 March 2024

Liouville-Bratu equation



models steady-state heat conduction plus an exponentially-growing exothermic reaction ... i.e. a warm block of explosive

Strong $\boxed{-\nabla^2 u - \lambda e^u = 0}$ ← seek $u(x,y) \in H_0^1(\Omega)$

Weak $\int_\Omega -(\nabla^2 u)v - \lambda e^u v = 0$

$\Leftrightarrow -\int_{\partial\Omega} \cancel{v}^{0} \nabla u \cdot \hat{n} + \int_\Omega \nabla u \cdot \nabla v - \lambda e^u v = 0$

$F(u) = \boxed{\int_\Omega \nabla u \cdot \nabla v - \lambda e^u v = 0}$ for all $v \in H_0^1(\Omega)$

Firedrake:

```
F = ( dot(grad(u), grad(v)) - lam*exp(u)*v )
                                    * dx
```

Newton method

$u^{(0)}$ initial guess of solution

$$F(u^{(k+1)}) = 0 \quad \leftarrow \text{want this}$$

$$F(u) \simeq F(u^{(k)}) + J_F(u^{(k)})(u - u^{(k)})$$

↑ linear approx. around $u^{(k)}$

Newton step equation:

$$J_F(u^{(k)})\, s = -F(u^{(k)})$$

basically: $As = b$

Firedrake computes this derivative (Jacobian) for you

update:

$$u^{(k+1)} = u^{(k)} + s$$

need to know: for nonlinear PDEs,

- set up weak form as usual

```
F = ...
```

- call `solve(F == 0, u, bcs, ...)` as usual ...
- but use these or similar:

```
'snes_type': 'newtonls',
'snes_converged_reason': None,
'snes_linesearch_type': 'basic',
'snes_rtol': 1e-8,
```

Firedrake defaults

- PETSc's SNES component will manage the Newton iteration, and you will get feedback on its success (esp. `SNES_CONVERGED_REASON`)
- what you *want* to see is a small number of Newton iterations, insensitive to resolution
- for nonlinear problems it *does* matter what is your initial iterate in u

demo!

bratu.py