How to Solve Your Horrible Equation using FEniCS

Jerome Troy

University of Delaware

April 29, 2020

What is FEniCS?

- Finite Element Computational Software
- Gives intuitive implementations for the finite element method to solve PDEs
- Written in Python - easy to understand
- Mirrors mathematical notation
- Github for codes used: `https://github.com/JeromeTroy/fenics-hgss.git`

A Brief Introduction to Finite Elements

- How it works:

$$Lu = f, \quad \text{some boundary conditions}$$

- Discretize domain, Ω, into some mesh
- Choose a *Test function* v such that $v = 0$ on the (Diriclet part of) boundary

$$\int_\Omega vLu\,dx = \int_\Omega vf\,dx$$

- Integrate by parts, boundary condition on v makes boundary terms vanish
- Reduces order of derivatives in equation through integration
- FEniCS wraps functional analysis parts of finite elements nicely so they are all taken care of

An Example

- Consider Laplace's Equation in a Circle of radius 1, with Dirichlet Boundary Condition:

$$\begin{aligned} \nabla^2 u = \partial_x^2 u + \partial_y^2 u = f(\mathbf{x}), \quad & \mathbf{x} \in \Omega := \left\{ \mathbf{x} \in \mathbb{R}^2 : 0 \leq ||\mathbf{x}|| < 1 \right\} \\ & u(\mathbf{x} \in \partial\Omega) = 1 \end{aligned} \tag{1}$$

- Let v be a piecewise $C^1(\Omega)$ test function such that $v(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \partial\Omega$
- Variational form

$$\int_\Omega v \nabla^2 u \, d\mathbf{x} = \int_{\partial\Omega} v \frac{\partial u}{\partial n} \, d\sigma - \int_\Omega \nabla v \cdot \nabla u \, d\mathbf{x} = \int_\Omega v f(\mathbf{x}) \, d\mathbf{x}$$

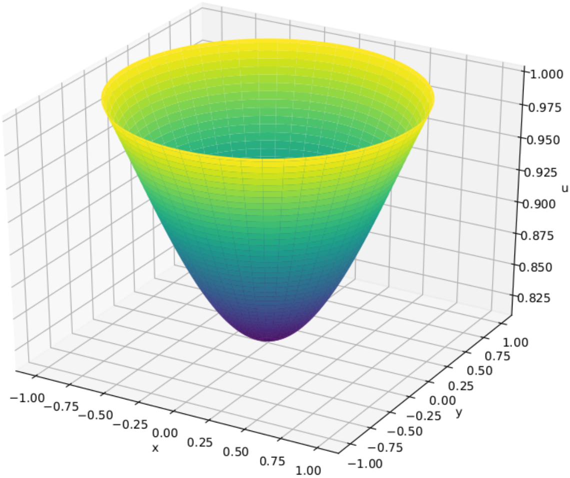
$$\text{BC on } v \implies A(u,v) := - \int_\Omega \nabla v \cdot \nabla u \, d\mathbf{x} = \int_\Omega v f(\mathbf{x}) \, d\mathbf{x} =: F(v; f)$$

The Code

```
# imports
from fenics import *
from mshr import *   # meshing utility
import matplotlib.pyplot as plt    # plotting

Ω = Circle(Point(0, 0), 1.0)

N = 100   # N is like 1 / Δx
mesh = generate_mesh(Ω, N)

V = FunctionSpace(mesh, "CG", 1)

u = TrialFunction(V)
v = TestFunction(V)

bc = DirichletBC(V, Constant(1.0), "on_boundary")

f_expr = Expression("1 - pow(x[0], 2) - pow(x[1], 2)", degree=1)
f = project(f, V)  # have to evaluate on function space

# variational form
A = -dot(grad(v), grad(u)) * dx
F = v * f * dx

# solve
u_sol = Function(V)
solve(A == F, u_sol, bc)
```

Solution using FEniCS

Example: $f(\mathbf{x}) = 1 - ||\mathbf{x}||_2^2$



Norm-2 Error: 0.002392

A Time Dependent Example

- Consider the heat equation:

$$\begin{aligned} \frac{\partial u}{\partial t} = \nabla \cdot (\kappa \nabla u), \quad & \mathbf{x} \in \Omega \\ & u(\mathbf{x}, t = 0) = u_0(\mathbf{x}) \end{aligned} \tag{2}$$

- Cylindrical Domain:

$$\Omega = \{x = (r\cos\theta, r\sin\theta, z) : 0 \leq \theta \leq 2\pi,\ 0 \leq r < 1,\ 0 < z < h\}$$

- Mixed boundary conditions:

$$u(r = 1, \theta, z, t) = u(r, \theta, z = 0, t) = 0, \quad \frac{\partial u}{\partial z}(r, \theta, z = h, t) = 0$$

Variational Form and Solution

- Split into many stationary problems: discretize time (Crank Nicolson time discretization)

$$t_n = n\Delta t, \quad u_n(\mathbf{x}) = u(\mathbf{x}, t_n)$$
$$\left.\frac{\partial u}{\partial t}\right|_{t_n} \approx \frac{u_{n+1} - u_n}{\Delta t} = \frac{1}{2}\left(\nabla \cdot (\kappa \nabla u_n) + \nabla \cdot (\kappa \nabla u_{n+1})\right)$$
$$\implies u_{n+1} - \frac{\Delta t}{2}\nabla \cdot (\kappa \nabla u_{n+1}) = u_n + \frac{\Delta t}{2}\nabla \cdot (\kappa \nabla u_n)$$

- Variational Form

$$\int_\Omega \left(v u_{n+1} + \frac{\Delta t}{2}\kappa \nabla v \cdot \nabla u_{n+1}\right) dx = \int_\Omega \left(v u_n - \frac{\Delta t}{2}\kappa \nabla v \cdot \nabla u_n\right) dx$$
$$A(v, u_{n+1}) = F(v, u_n)$$

- Solve $A(v, u_{n+1}) = F(v, u_n)$ subject to the boundary conditions for $n = 1, ..., N$

Heat Equation Solution

Linear Heat Equation Solution

Nonlinear Heat Equation

- $\kappa \to \kappa(u) = 1 - \frac{1-k}{2}\left[1 - \tanh\left(\frac{u-\tau}{\eta}\right)\right]$
- $0 \le k \le 1, \quad 0 \le \tau \le 1, \quad \eta > 0$
- Mimics a phase change where $\kappa = 1$ for high u, then $\kappa = \frac{1+k}{2} \le 1$ for low u
- η controls rate of switching from high to low κ, τ is phase change temperature
- Variational Form:

$$F(v, u_n; \Delta t) = \int_\Omega \left(v u_n - \frac{\Delta t}{2}\kappa(u_n)\nabla v \cdot \nabla u \right) dx$$

$$A(v, u_{n+1}; \Delta t) = \int_\Omega \left(v u_{n+1} + \frac{\Delta t}{2}\kappa(u_{n+1})\nabla v \cdot \nabla u_{n+1} \right) dx$$

- Solve $A(v, u_{n+1}) = F(v, u_n)$ subject to same boundary conditions

Comparison of Linear and Nonlinear Heat Equations

Comparison Solutions

A Nonlinear (Horrible) Example

- The catenary problem

$$\min_{u} \int_{-1}^{1} u\sqrt{1+(\epsilon u')^2}\,dx, \quad \text{subject to} \quad \int_{-1}^{1} \sqrt{1+(\epsilon u')^2}\,dx = \ell \tag{3}$$

With $u(\pm 1) = 0$.

- FEniCS applied to Lagrange Multiplier Problem

$$\min_{u,\lambda} \mathcal{F}(u;\lambda) = \int_{-1}^{1} u\sqrt{1+(\epsilon u')^2}\,dx - \lambda \int_{-1}^{1} \left[\frac{\ell}{2} - \sqrt{1+(\epsilon u')^2}\right] dx \tag{4}$$

- Note $\mathcal{F} : C^1[-1,1] \times \mathbb{R} \to \mathbb{R}$
- Let $X := C^1[-1,1] \times \mathbb{R}$
- For simplicity, let

$$\frac{\mathrm{d}s}{\mathrm{d}x} = \sqrt{1+(\epsilon u')^2}$$

Horrible Problem Made Easy

- Some notation to simplify
  - $X := C^1[-1,1] \times \mathbb{R}$ mixed space
  - $y \in X$, $y = (u, \lambda)$
- Minimize $\mathcal{F}(y) = \mathcal{F}((u,\lambda)) = \int_{-1}^{1} u \frac{\mathrm{d}s}{\mathrm{d}x} - \lambda \left[\frac{\ell}{2} - \frac{\mathrm{d}s}{\mathrm{d}x} \right] dx$

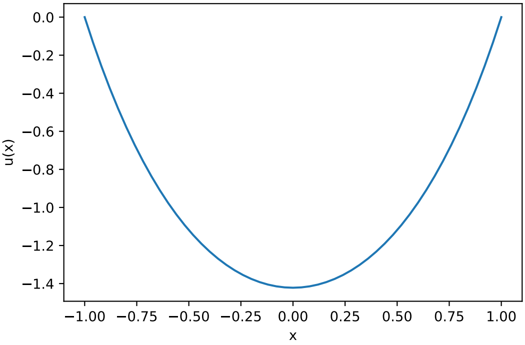


Figure: Solution to the Catenary Problem using FEniCS

References

Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells.
The fenics project version 1.5.
Archive of Numerical Software, 3(100), 2015.

Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al.
Automated Solution of Differential Equations by the Finite Element Method.
Springer, 2012.

https://github.com/JeromeTroy/fenics-hgss.git