

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

$$\int_{\Omega} v Lu \, dx = \int_{\Omega} v f \, dx$$

- 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:

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

- 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

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

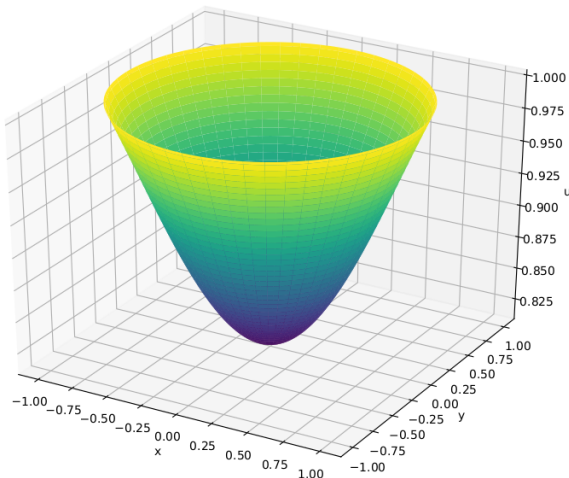
$$\text{BC on } v \implies A(u, v) := - \int_{\Omega} \nabla v \cdot \nabla u \, d\mathbf{x} = \int_{\Omega} v f(\mathbf{x}) \, d\mathbf{x} =: F(v; f)$$

# The Code

```
1  # imports
2  from fenics import *
3  from mshr import *   # meshing utility
4  import matplotlib.pyplot as plt   # plotting
5
6   $\Omega$  = Circle(Point(0, 0), 1.0)
7
8  N = 100   # N is like 1 /  $\Delta x$ 
9  mesh = generate_mesh( $\Omega$ , N)
10
11 V = FunctionSpace(mesh, "CG", 1)
12
13 u = TrialFunction(V)
14 v = TestFunction(V)
15
16 bc = DirichletBC(V, Constant(1.0), "on_boundary")
17
18 f_expr = Expression("1 - pow(x[0], 2) - pow(x[1], 2)", degree=1)
19 f = project(f, V)   # have to evaluate on function space
20
21 # variational form
22 A = -dot(grad(v), grad(u)) * dx
23 F = v * f * dx
24
25 # solve
26 u_sol = Function(V)
27 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)

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

- Variational Form

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

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

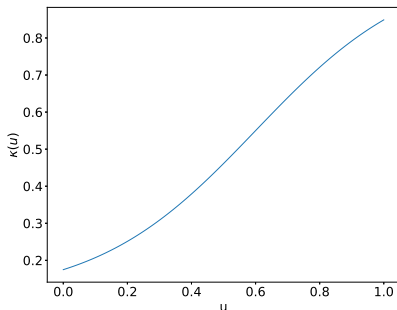


# Heat Equation Solution

## Linear Heat Equation Solution

# Nonlinear Heat Equation

- $\kappa \rightarrow \kappa(u) = 1 - \frac{1-k}{2} \left[ 1 - \tanh \left( \frac{u-\tau}{\eta} \right) \right]$
- Mimics a phase change where  $\kappa = 1$  for high  $u$ , then low for low  $u$
- Variational Form:



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

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

# Comparison of Linear and Nonlinear Heat Equations

Comparison Solutions

# A Nonlinear (Horrible) Example

- The catenary problem

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

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

- FEniCS applied to Lagrange Multiplier Problem

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

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

$$\frac{ds}{dx} = \sqrt{1 + (\epsilon u')^2}$$

# 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  $\mathcal{F}(y) = \mathcal{F}((u, \lambda)) = \int_{-1}^1 u \frac{ds}{dx} - \lambda \left[ \frac{\ell}{2} - \frac{ds}{dx} \right] dx$

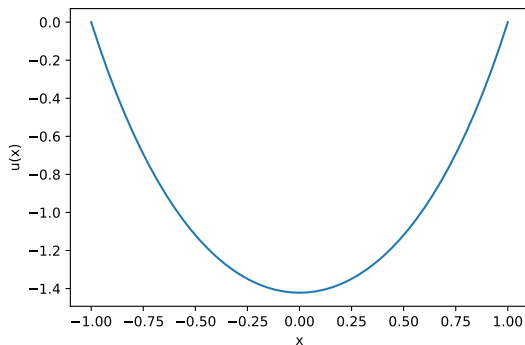


Figure: Solution to the Catenary Problem using FEniCS

# Conclusions

- FEniCS provides an intuitive implementation for finite elements
- FEniCS can be overkill for simple problems - it has a lot of overhead
- Can solve complicated problems with relative ease
- Many more examples on the FEniCS website and book (free)

# 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>