IN5270 Project 2

Jens Due

November 2019

1 Method

The goal for this project is to solve the nonlinear diffusion equation

$$\rho u_t = \nabla \cdot (\alpha(u)\nabla u) + f(\mathbf{x}, t),$$

with the initial condition $u(\mathbf{x}, 0) = I(\mathbf{x})$ and boundary condition $\partial u/\partial n = 0$, where the coefficient ρ is constant and α is a known function of u.

1.1 Backward Euler finite difference method

First off we would need to construct an implicit scheme by using Backward Euler in the time direction. Thus, we need to estimate the exact value of u as $u^n = \sum_{j=0}^{N} c_j \psi_j(\mathbf{x})$.

$$\begin{aligned}
\rho u_t &= \nabla \cdot (\alpha(u)\nabla u) + f(\mathbf{x}, t) \\
\rho \frac{u^n - u^{n-1}}{\Delta t} &= \nabla \cdot (\alpha(u^n)\nabla u^n) + f \\
u^n &= \frac{\Delta t}{\rho}\Big[\nabla \cdot (\alpha(u^n)\nabla u^n) + f\Big] + u^{n-1}
\end{aligned}$$

Thereafter, we need to find the variational formulation for each time step. This is done by requiring that the residual of the equation is orthogonal to all functions $v \in V$, meaning $\int_\Omega R \cdot v d\mathbf{x} = 0$. The residual is found by rearranging all terms to one side

$$\begin{aligned}
R = \frac{\Delta t}{\rho}\Big[\nabla \cdot (\alpha(u^n)\nabla u^n) + f\Big] + u^{n-1} - u^n = 0 \\
\int_\Omega \Big(\frac{\Delta t}{\rho}\Big[\nabla \cdot (\alpha(u^n)\nabla u^n) + f\Big] + u^{n-1} - u^n\Big) v \cdot d\mathbf{x} = 0 \\
\int_\Omega \Big(\frac{\Delta t}{\rho}\Big[\nabla(\alpha(u^n)\nabla u^n)v + fv\Big] + u^{n-1}v - u^n v\Big) \cdot d\mathbf{x} = 0
\end{aligned}$$

To get rid of second derivatives in the first term, we will use integration by parts.

$$\int_\Omega \nabla(\alpha \nabla u^n) v d\mathbf{x} = \Big[\alpha \nabla u^n v\Big]_{d\Omega} - \int_\Omega \alpha \nabla u^n \nabla v \cdot d\mathbf{x}$$

Due to the Neumann condition $\partial u/\partial n = 0$, the first term disappears.

$$\int_\Omega \nabla(\alpha\nabla u^n)v d\mathbf{x} = -\int_\Omega \alpha\nabla u^n\nabla v \cdot d\mathbf{x}$$

The complete variational form is then

$$\int_\Omega \Big(\frac{\Delta t}{\rho}\Big[-\alpha\nabla u^n\nabla v + fv\Big] + u^{n-1}v - u^n v\Big)\cdot d\mathbf{x} = 0$$

For the initial condition $u(\mathbf{x},0) = I(\mathbf{x})$ we have for $n = 1$ that

$$\int_\Omega \Big(I(\mathbf{x})v - u^1 v - \frac{\Delta t}{\rho}(\alpha\nabla u^1\nabla v) + \frac{\Delta t}{\rho}f(\mathbf{x},0)v\Big)d\mathbf{x} = 0$$

1.2 Picard iteration

There is still an issue needing to be adressed, the equation is nonlinear, due to the term $\alpha(u)$. We will be using Picard iteration to linearize our problem at the PDE level. A general Picard iteration is performed by swapping out the dependency of the nonlinear term from the current solution u, to an approximation u^k which is known. u^k is found by performing a previous Picard iteration. Our PDE is currently

$$\rho u_t = \nabla\cdot(\alpha(u)\nabla u) + f(\mathbf{x},t),$$

and now we set $\alpha(u) \approx \alpha(u^k)$, effectively nonlinearizing the equation.

$$\rho u_t = \nabla\cdot(\alpha(u^k)\nabla u) + f(\mathbf{x},t),$$

1.3 One picard iteration

For one single picard iteration, it is standard procedure to use the solution of the previous time step as the initial approximation. $u^k = u^- = u^{n-1}$ Our variational form will then be

$$\int_\Omega \rho u^n v dx - \int_\Omega u^{n-1}v dx + \Delta\int_\Omega \alpha(u^{n-1})\nabla u^n\nabla v dx - \Delta t\int_\Omega f(\mathbf{x},t_n)v dx = 0$$

This has been implemented in the program.

1.4 First verification

For reproducing a constant solution, we achieved a max absolute difference of $1.55\cdot 10^{-15}$.

1.5 Second verification

h	E/h
0.0100	$1.7885 \cdot 10^{-3}$
0.0044	$1.6318 \cdot 10^{-3}$
0.0025	$1.54133 \cdot 10^{-3}$
0.0011	$1.5179 \cdot 10^{-3}$
0.0004	$1.5025 \cdot 10^{-3}$

Table 1: Table of step length and error as the mesh increases without a source term.

1.6 Third verification

h	E/h
0.0100	$1.5794 \cdot 10^{-3}$
0.0025	$1.5589 \cdot 10^{-3}$
0.0011	$1.5459 \cdot 10^{-3}$
0.0004	$1.5393 \cdot 10^{-3}$
0.0001	$1.5365 \cdot 10^{-3}$

Table 2: Table of step length and error as the mesh increases with a source term.

1.7 Different sources of numerical errors in the FEniCS program

FEniCS is a bit of a mystery, but numerical errors do arise. The biggest source of errors will be in the approximation of the derivative from Backward Euler as well as the picard approximation. Another error can be due to machine precision, but this has a smaller impact than the former.