Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4183 Opt. II
Spring 2016

Exercise set 4^{num}

The idea is to apply a very simple numerical PDE discretization technique for olving a linear-quadratic control problem for Laplace equation without control constraints.

Thus we are trying to solve the coupled system of PDEs (see page 71 in [Tr]):

$$\begin{aligned} -\Delta y &= -\lambda^{-1}\beta^2 p & \qquad -\Delta p &= y - y_\Omega \\ y|_\Gamma &= 0 & p|_\Gamma &= 0, \end{aligned}$$

where $\lambda > 0$, $\beta \in L^\infty(\Omega)$, $y_\Omega \in L^2(\Omega)$ are given. After the optimal state and the corresponding adjoint state have been determined, one computes the control as $u = -\lambda^{-1}\beta p$.

1 Find the weak formulation of the coupled PDE system.

2 Let $\Omega = (0,1)^2 \subset \mathbb{R}^2$. Solve the coupled system of PDEs using some numerical discretization method (see below) and verify the correctness of your implementation. For example, take $\lambda = 1$, $\beta = 1$, and take $y \in H_0^1(\Omega)$ such that $\Delta y \in H_0^1(\Omega)$ (e.g., $y = [x_1x_2(1-x_1)(1-x_2)]^3$). Then from the first PDE we can determine $p \in H_0^1(\Omega)$, and from the second y_Ω. Provide these parameters as input to your numerical implementation and check that it does converge to the analytical soutions y and p.

Hint: it is probably a good idea to use some numerical differentiation software to compute y_Ω.

Discussion of possible numerical methods

1. One of the simplest possible choices is a 5-point finite difference discretization of Laplace operator on a structured grid, which is easily generalizable to the present case. See for example `https://en.wikipedia.org/wiki/Five-point_stencil#Two_dimensions`
2. Another option is to decompose Ω into polygons (cells) and approximate the solution to the system using piecewise-constant functions (constant over each cell). The discretization of Laplace equation implementing this strategy is described briefly in `FVMLaplace.pdf`, and the Matlab code (most of it is a polygonal mesher) is available in `FVMLaplace.zip`. The Laplace discretization is easily generalizable to the coupled system at stake.

3. Another option is to use the finite element method (FEM). I recommend using a high-level open source library FEniCS (`http://fenicsproject.org`). The full Python code for solving the system of optimality conditions stated above is found in `ex4_num.py`