Norwegian University of Science
and Technology
Department of Mathematical
Sciences

TMA4183 Opt. II
Spring 2017

Exercise set 4

Reading material: Sections 2.9, 2.12 from [Tröltsch].

Download and install FENICS and Paraview (for visualization). Study the codes `laplace00.py - laplace06.py` which show how to solve Laplace equation with Dirichlet boundary conditions, evaluate the objective function, differentiate it (and check derivatives using finite differences), and finally solve unconstrained and bound constrained problems (the latter using the conditional gradient method).

1 Consider the following slight modification of the distributed control problem we have been discussing:

$$\min \; J(y,u) = \frac{1}{2}\|y - y_\Omega\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u\|^2_{L^2(\Omega)},$$

where

$$u \in U_{\text{adm}} = \{\, u_a(x) \le u(x) \le u_b(x) \,\}$$
$$-\Delta y(x) = \beta(x)u(x) + f(x), \qquad x \in \Omega$$
$$y(x) = y_0(x), \qquad x \in \partial\Omega.$$

Thus the only difference is that the state problem contains an additional term $f(x)$ in the right hand side; we assume that $f(x)$ is given and independent from y, u.

a) State the first order optimality conditions for this problem.

b) Define a test example for this problem.

You can start by selecting u_a, u_b, β, $\bar{u}$, and $\bar{y}$. Based on these one can compute y_0 and f such that the state equations are satisfied. Then one should chose $\bar{p}$ such that the optimality condition $L'_u(\bar{y},\bar{u},\bar{p})[u-\bar{u}] \ge 0$, $\forall u \in U_{\text{adm}}$ is satisfied (here L is the Lagrangian function).

Finally, one can compute y_Ω such that the adjoint problem is solved by $\bar{p}$.

See Section 2.9 in [Tr] for inspiration and details.

c) Modify the conditional gradient algorithm in `laplace06.py` as necessary to solve the modified control problem. Check whether the algorithm converges to the expected solution, and whether the accuracy improves as the mesh is refined (to do this you need to make sure that the stopping criterion is "strong enough" so that the algorithm does not stop prematurely before converging accurately enough to a solution).

2 **a)** Starting with the code for the conditional gradient algorithm in the previous exercise, implement a *p*rojected gradient algorithm; see Section 2.12.2 in [Tr]. Note that the step length in the projected gradient algorithm cannot be determind analytically in general. Use the bisection algorithm (also known as the backtracking linesearch), see p. 95 in [Tr]. Accept the step size when it satisfies the sufficient decrease condition, i.e. when the *actual* decrease in the objective function is at least a small fraction of than what is predicted by a first order model:

$$f(\mathbb{P}_{[u_a,u_b]}(u_n + sv_n)) \leq f(u_n) + c_n f'(u_n)[\mathbb{P}_{[u_a,u_b]}(u_n + sv_n) - u_n],$$

where c_n is a small constant (one typically uses $c_n \approx 10^{-3}$ or similar).
Verify your implementation using the test example you have constructed previously.

b) * Implement the primal-dual active set algorithm for the same problem, see section 2.12.4 in [Tr]. Note that the derivation there assumes $\beta = 1$, but one does not have to do this!
When solving the system (2.97) [Tr] it is best to exclude u from it. What remains is a coupled system of PDEs for (y, p), which can be solved in FENICS similarly to the unconstrained case, see `laplace05.py` .
Verify your implementation using the test example you have constructed previously.

3 * Consider the optimal stationary boundary temperature problem:

$$\begin{aligned}
&\min\ J(y,u) = \frac{1}{2}\|y - y_\Omega\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u\|^2_{L^2(\partial\Omega)},\\
&\text{where}\\
&u \in U_{\text{adm}} = \{\, u_a(x) \leq u(x) \leq u_b(x), \quad x \in \partial\Omega \,\}\\
&-\Delta y(x) = \beta(x)u(x) + f(x), \qquad x \in \Omega\\
&\partial_\nu y + \alpha y(x) = \alpha u(x) + g(x), \qquad x \in \partial\Omega,
\end{aligned}$$

see section 2.8.4 in [Tr].

a) Derive (or find in [Tr]) the weak formulation of the state problem and implement in in FENICS - follow the example `laplace00.m` . You can ignore the control u in this part of the exercise. Note that boundary integrals in FENICS are denoted with `*ds` and not `*dx` .

b) Compute the directional derivative of the reduced objective function using adjoint method. Implement this computation in FENICS, and verify your implementation with finite difference calculations.
Note: the control space can be defined as

```
mesh = UnitSquareMesh(N,N)
boundary_mesh = BoundaryMesh(mesh, 'exterior')
U = FunctionSpace(boundary_mesh,'DG',0)
```

c) Derive a test case for this problem. Adapt your implementation of the projected gradient method from the previous exercise to this boundary control problem. Verify your implementation using the test case.