Fall 2015: Computational and Variational Methods for Inverse Problems CSE 397/GEO 391/ME 397/ORI 397 Assignment 3 (due 2 November 2015)

The problems below require a mix of paper-and-pencil work and FEniCS implementation. Example implementations in FEniCS (as used for Problems 2 and 3 below) can be found on the class website, `http://users.ices.utexas.edu/~omar/inverse_problems/index.html`. Please hand in printouts of your FEniCS implementations together with the results.

1. The problem of removing noise from an image without blurring sharp edges can be formulated as an infinite-dimensional minimization problem. Given a possibly noisy image $d(x,y)$ defined within a square domain Ω, we would like to find the image $u(x,y)$ that is closest in the L_2 sense, i.e. we want to minimize

$$\mathcal{F}_{LS} := \frac{1}{2}\int_\Omega (u-u_0)^2 \, d\boldsymbol{x},$$

while also removing noise. This noise is assumed to comprise very "rough" (i.e. highly oscillatory) components of the image. This latter goal can be incorporated as an additional term in the objective, in the form of a penalty, i.e.,

$$\mathcal{R}_{TN} := \frac{1}{2}\int_\Omega k(\boldsymbol{x}) \nabla u \cdot \nabla u \, d\boldsymbol{x},$$

where $k(\boldsymbol{x})$ acts as a "diffusion" coefficient that controls how strongly we impose the penalty, i.e. how much smoothing occurs. Unfortunately, if there are sharp edges in the image, this so-called *Tikhonov (TN) regularization* will blur them. Instead, in these cases we prefer the so-called *total variation (TV) regularization*,

$$\mathcal{R}_{TV} := \int_\Omega k(\boldsymbol{x}) (\nabla u \cdot \nabla u)^{\frac{1}{2}} \, d\boldsymbol{x}$$

where (we will see that) taking the square root is the key to preserving edges. Since $\mathcal{R}_{TV}$ is not differentiable when $\nabla u = \boldsymbol{0}$, it is usually modified to include a positive parameter ε as follows:

$$\mathcal{R}^{\varepsilon}_{TV} := \int_\Omega k(\boldsymbol{x}) (\nabla u \cdot \nabla u + \varepsilon)^{\frac{1}{2}} \, d\boldsymbol{x}.$$

We wish to study the performance of the two denoising functionals $\mathcal{F}_{TN}$ and $\mathcal{F}^{\varepsilon}_{TV}$, where

$$\mathcal{F}_{TN} := \mathcal{F}_{LS} + \mathcal{R}_{TN}$$

and

$$\mathcal{F}^{\varepsilon}_{TV} := \mathcal{F}_{LS} + \mathcal{R}^{\varepsilon}_{TV}.$$

We prescribe the homogeneous Neumann condition $\nabla u \cdot \boldsymbol{n} = 0$ on the four sides of the square domain, which amounts to assuming that the image intensity does not change normal to the boundary of the image.

(a) For both $\mathcal{F}_{TN}$ and $\mathcal{F}_{TV}^{\varepsilon}$, derive the first-order necessary condition for optimality using calculus of variations, in both weak form and strong form. Use $\hat{u}$ to represent the variation of u.

(b) Show that when $\boldsymbol{\nabla} u$ is zero, $\mathcal{R}_{TV}$ is not differentiable, but $\mathcal{R}_{TV}^{\varepsilon}$ is.

(c) For both $\mathcal{F}_{TN}$ and $\mathcal{F}_{TV}^{\varepsilon}$, derive the infinite-dimensional Newton step, in both weak and strong form. For consistency of notation, please use $\tilde{u}$ as the differential of u (i.e. the Newton step). The strong form of the second variation of $\mathcal{F}_{TV}^{\varepsilon}$ will give an anisotropic diffusion operator of the form $-\text{div}(\boldsymbol{A}(u)\nabla\tilde{u})$, where $\boldsymbol{A}(u)$ is an anisotropic tensor that plays the role of the diffusion coefficient.[1] (In contrast, you can think of the second variation of $\mathcal{F}_{TN}$ giving an *isotropic* diffusion operator, i.e. with $\boldsymbol{A} = \alpha\boldsymbol{I}$ for some α.)

(d) Derive expressions for the two eigenvalues and corresponding eigenvectors of $\boldsymbol{A}$. Based on these expressions, give an explanation of why $\mathcal{F}_{TV}^{\varepsilon}$ is effective at preserving sharp edges in the image, while $\mathcal{F}_{TN}$ is not. Consider a single Newton step for this argument.

(e) Show that for large enough ε, $\mathcal{R}_{TV}^{\varepsilon}$ behaves like $\mathcal{R}_{TN}$, and for $\varepsilon = 0$, the Hessian of $\mathcal{R}_{TV}^{\varepsilon}$ is singular. This suggests that ε should be chosen small enough that edge preservation is not lost, but not too small that ill-conditioning occurs.

2. An anisotropic Poisson problem in a two-dimensional domain Ω is given by the strong form

$$-\nabla \cdot (\boldsymbol{A}\nabla u) = f \qquad \text{in } \Omega, \tag{1a}$$

$$u = u_0 \qquad \text{on } \Gamma, \tag{1b}$$

where the conductivity tensor $\boldsymbol{A}(\boldsymbol{x}) \in \mathbb{R}^{2\times 2}$ is assumed to be symmetric and positive definite for all $\boldsymbol{x}$, $f(\boldsymbol{x})$ is a given distributed source, and $u_0(\boldsymbol{x})$ is the source on the boundary Γ.[2]

(a) Derive the variational/weak form corresponding to the above problem, and give the energy functional that is minimized by the solution u of (1).

(b) Solve problem (1) in FEniCS using quadratic finite elements. Choose Ω to be a disc with radius 1 around the origin and take the source terms to be

$$f = \exp(-100(x^2 + y^2)) \quad \text{and} \quad u_0 = 0.$$

Use conductivity tensors $\boldsymbol{A}(\boldsymbol{x})$ given by

$$\boldsymbol{A}_1 = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \text{ and } \boldsymbol{A}_2 = \begin{pmatrix} 1 & -5 \\ -5 & 100 \end{pmatrix}$$

[1] Hint: For vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^n$, note the identity $(\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c} = (\boldsymbol{c}\boldsymbol{a}^T)\boldsymbol{b}$, where $\boldsymbol{a}\cdot\boldsymbol{b} \in \mathbb{R}$ is the inner product and $\boldsymbol{c}\boldsymbol{a}^T \in \mathbb{R}^{n\times n}$ is a matrix of rank one.

[2] One interpretation of the boundary value problem (1) is that it describes the steady state conduction of heat in a solid body. In this case, u is the temperature, $\boldsymbol{A}$ is the thermal conductivity, f is the distributed heat source, and the temperature on the boundary Γ is maintained at u_0.

and compare the results obtained using $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in (1). A mesh for the unit circle is provided in the file `circle.xml`. You can load the mesh into FEniCS using the command

```
mesh = Mesh("circle.xml")
```

To define the conductivity tensor $\boldsymbol{A}(\boldsymbol{x})$ use the commands

```
A1 = Expression((("10", "0"),("0", "10")))
A2 = Expression((("1", "-5"),("-5", "100")))
```

3. Implement the image denoising method from Problem 1 above in FEniCS using Tikhonov (TN) and total variation (TV) regularizations. To this end, set $k(\boldsymbol{x}) = \alpha$ with small $\alpha > 0$ in $\mathcal{R}_{TV}$ and $\mathcal{R}_{TN}$, choose small $\varepsilon > 0$, and take a homogeneous Neumann boundary condition for u (i.e., $\nabla u \cdot \boldsymbol{n} = 0$). The file `tntv.py` contains some lines to start up the implementation (definition of the mesh, finite element space, and an expression to evaluate the true image and the noisy image at each point of the mesh).

   (a) Solve the denoising inverse problem using TN regularization. Since for TN regularization, the gradient is linear in u, you can use FEniCS's linear solver `solve`. Choose an $\alpha > 0$ such that you obtain a reasonable reconstruction, i.e., a reconstruction that removes noise from the image but does not overly smooth the image.[3]

   (b) Solve the denoising inverse problem defined above using TV regularization. Since in this case the gradient is nonlinear in u, use the `InexactNewtonCG` nonlinear solver provided in the file `unconstrainedMinimization.py`. This solver uses the inexact Newton CG method with Eisenstat-Walker early termination condition and Armijo-based backtracking line search. It uses FEniCS's built-in CG algorithm, which does not support early termination due to detection of negative curvature; this is appropriate since the image denoising objective functionals for both TN and TV result in positive definite Hessians.[4] Find an appropriate value for α.[5] You will have to increase the default number of nonlinear iterations in `InexactNewtonCG`.[6] How does the number of nonlinear iterations behave for decreasing ε (e.g., from 10 to 10^{-4})? Try to explain this behavior.[7]

[3] Either experiment manually with a few values for α or use the L-curve criterion.

[4] See the file `energyMinimization.py` for an example of how to use this nonlinear solver. FEniCS's built-in nonlinear solver is not appropriate for this problem, since it does not know about the underlying optimization cost functional. On the other hand, implementing our own nonlinear solver allows us to globalize Newton's method via an Armijo line search based on knowledge of the cost functional.

[5] Try out a few different values for α. Using the L-curve or Morozov criterion is problematic here, since unlike TN regularization, TV is not quadratic, which makes the automatic choice of α harder.

[6] Typing "`help(InexactNewtonCG)`" will show you solver options. You should set the relative tolerance `rel_tolerance` to 10^{-5} and increase the value of `max_iter`, which defaults to 20.

[7] For small values of ε, there are more efficient methods for solving TV-regularized inverse problems than the basic Newton method we use here; in particular, so-called primal-dual Newton methods are preferred (see *T.F. Chan, G.H. Golub, and P. Mulet, A nonlinear primal-dual method for total variation-based image restoration, SIAM Journal on Scientific Computing, 20(6):1964–1977, 1999*). The efficient solution of TV-regularized inverse problems is still an active field of research.

(c) Compare the denoised images obtained with TN and TV regularizations, using the insight derived from your answers to Problem 1.