12-759: Computational Optimization of Systems Governed by PDEs
Assignment 3: Image Denoising – Computation
Due 12 November 2003

In the second assignment, you studied the problem of removing noise from an image without blurring sharp edges. The problem setup is repeated here for convenience. Recall that this can be formulated as an infinite-dimensional minimization problem. Given a possibly noisy initial image $u_0(x, y)$, we would like to find the improved 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, which is assumed to comprise very "rough" components of the image. This latter goal can be incorporated as an additional "penalty" term in the objective, either as the Tikhonov regularization

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

or the total variation (TV) regularization

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

where β is a constant that controls how strongly we impose the penalty, and ε is a small positive constant added to insure differentiability of $\mathcal{R}_{TV}^{\varepsilon}$ wherever $\nabla u = \boldsymbol{0}$.

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

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

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

with the homogeneous Neumann boundary condition $\nabla u \cdot \boldsymbol{n} = 0$,

For the noisy image, let's start with the following "baseline" image:

$$u^*(x, y) = \frac{1}{2}[1 + \tanh(\alpha\phi)], \qquad \text{in} \quad \Omega = [0, 6] \times [0, 6],$$

where $\alpha = 100$ and

$$\phi(x, y) = -94 + 61x - 41x^2 + 11x^3 - x^4 + 96y - 52y^2 + 12y^3 - y^4$$

Now let's add some "noise" to u^* to obtain the noisy image u_0,

$$u_0(x, y) = u^*(x, y) + \eta\left(\sin\frac{\pi x}{2h} + \sin\frac{\pi y}{2h}\right)$$

where h is the pixel size (which we will take to be the same as the mesh size) and η is the "noise level." The cosine expression doesn't really represent noise, but it does have a wavelength of $4h$ so it simulates pixel-level artifacts.

1. Using the minimal surface code that I demoed in class as a guide (which you can download from the class website; look for `ms.cpp` under *Handouts*), and the expressions you derived on the last assignment for the Newton step, implement a Newton solver for this problem. Test your code on the above problem for a noise level $\eta = 0.1$, on a 64×64 mesh of linear elements, with $\varepsilon = 0.001$. You will have to experiment to determine the best value of the regularization constant β.

2. Study the effect of a change in:

   - the mesh size
   - the regularization parameter β
   - ε
   - the noise level η
   - the type of regularizer (either TV or Tikhonov)

   on both the quality of the inverted solution, as well as the convergence of Newton's method.

3. Implement a so-called *Picard* method to solve this problem, in which the Hessian of the TV regularization functional is approximated by dropping the tensor part of diffusion coefficient $\mathcal{A}$ (see previous assignment). This generally leads to improved numerical behavior, since the approximate Hessian is much better conditioned than the exact one. Note that you cannot rely on Sundance's linearization to create the Newton equation for you, since it will generate the exact Hessian; you must do that manually.