BME 493/593, Spring 2019:
Computational Methods for Imaging Science
Assignment 4 (due April 8, 2019)

Instructions: The problems below require a mix of paper-and-pencil work and Python implementation using the Jetstream cloud resources[1]. Remember that, as stated in the syllabus, you should both

1. Submit a hard copy of your solutions (which will be collected on Monday, April 8 **before** the class starts), **and**
2. Upload a digital copy of your solutions on Canvas by 9:59 AM on Monday, April 8.

Please include printouts of the python code/notebooks together with the generated results and figures in the hard copy of your solutions.
The digital copy uploaded on Canvas must be a scan (or *readable* cellphone picture) of the hard copy that you handed in at the beginning of class. The instructors will grade only the hard copy of your solutions. The digital copy uploaded on Canvas only serves as proof that the assignment was turned in on time. If you are late or miss class on the day the assignment is due, uploading your solution on Canvas before the due date ensures that your assignment is considered on time.

To access the Jetstream cloud computing resources follow the link `http://uvilla.github.io/cmis_labs/cloud.html` and use your wustl e-mail address (all lowercase and without `@wustl.edu`) as username and your student id as password.

For this assignment you will need the following files (that you can find inside the `cmis_lab/Assignment4` folder on the Jetstream cloud resources):

- `Problem1.ipynb`: A jupyter notebook to solve the anisotropic reaction-diffusion boundary value problem in **Problem 1**.
- `ImageDenoising_TV.ipynb`: The jupyter notebook to solve the Total Variation denoising problem using the primal formulation (as shown in class on March 25).
- `PrimalDual_TV.ipynb`: The jupyter notebook to solve the Total Variation denoising problem using the primal-dual formulation (as shown in class on March 27).
- `circles.mat`: The ground truth image to solve **Problem 2** and **Problem 3**.

Problem 1. An anisotropic diffusion-reaction problem in a two-dimensional domain Ω is given by the strong form

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

$$A(x)\nabla m \cdot \boldsymbol{n} = 0 \qquad \text{on } \partial\Omega, \tag{1b}$$

[1] These resources were awarded to us by the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Fundation.

where $A(x) \in \mathbb{R}^{2\times 2}$ is assumed to be symmetric and positive definite for every x, f is a given force, and $\boldsymbol{n} \in \mathbb{R}^2$ is the unit outward normal vector to the boundary $\partial\Omega$.

a) Derive the variational form corresponding to the boundary value problem (1).

b) Run the provided Jupyter notebook `Problem1.ipynb`. This code solves problem (1) in FEniCS using quadratic finite elements on a square domain $\Omega = [-1, 1]^2$. The forcing term f is given by

$$f = \exp(-100(x^2 + y^2)),$$

and the diffusion matrices $A(x)$ is either

$$A_1 = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \text{ or } A_2 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 10 \end{pmatrix}.$$

c) Compare the results obtained with A_1 and A_2 in (1).

Problem 2. 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 rectangular domain Ω, we would like to find the image $m(x, y)$ that is closest in the L_2 sense, i.e. we want to minimize

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

while also removing noise, which is assumed to comprise very *rough* components of the image. This latter goal can be incorporated as an additional term in the objective, in the form of a penalty.

Specifically, we consider a differentiable approximation to the total variation functional given by

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

Then the solution $m^\star$ of the denoising problem is then the minimizer of the functional $J(m)$ defined by

$$J(m) = \frac{1}{2}\int_\Omega (m - d)^2 \, d\boldsymbol{x} + \alpha \int_\Omega (\nabla m \cdot \nabla m + \beta)^{\frac{1}{2}} \, d\boldsymbol{x}.$$

a) Derive the first-order necessary condition for optimality using calculus of variations, in both variational form and strong form. Use $\tilde{m}$ to represent the variation of m. Check your derivation against the expression provided in class and in the notebook `ImageDenoising_TV.ipynb`.

b) Derive the infinite-dimensional Newton step, in both variational and strong form. For consistency of notation, please use $\hat{m}$ as the differential of m (i.e. the Newton direction). Check your derivation against the expression provided in class and in the notebook `ImageDenoising_TV.ipynb`.

c) Consider the symmetric positive definite anysotropic diffusion coefficient

$$A(\boldsymbol{x}) := \frac{1}{\sqrt{\nabla m \cdot \nabla m + \beta}} \left(I - \frac{\nabla m \cdot \nabla m}{\nabla m \cdot \nabla m + \beta} \right),$$

which appears in the strong form of the Newton step. Derive expressions for the two eigenvalues and corresponding eigenvectors of $A(\boldsymbol{x})$. Based on these expressions and on what you observed in **Problem 1**, give an explanation of why $\mathcal{R}_{TV}^{\beta}$ is effective at preserving sharp edges in the image, while Tikhonov regularization is not. Consider a single Newton step for this argument.

d) Show analytically that for large enough β, $\mathcal{R}_{TV}^{\beta}$ behaves like Tikhonov regularization, and for $\beta = 0$, the Hessian of $\mathcal{R}_{TV}^{\beta}$ is singular. This suggests that β should be chosen small enough that edge preservation is not lost, but not too small that ill-conditioning occurs.

e) Use the code provided in `ImageDenoising_TV.ipynb` to solve the denoising problem with $\alpha = 10^{-3}$ and different values of β. For $\beta = 10, 1, 0.1, 0.01, 10^{-3}, 10^{-4}$, show the reconstructed images and report the number of Newton iterations. How does the number of Newton iterations behave for decreasing β? Explain this behavior based on your answer to Question (d).

Problem 3. For small values of β, there are more efficient methods for solving TV-regularized inverse problems than the basic Newton method we used in **Problem 2**. In particular, the notebook `PDNewton.ipynb` implements the so-called primal-dual Newton methods (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).

a) Set $\beta = 10^{-4}$ in the code, and use the L-curve criterion to find a reasonable choice of α.

b) Set $\alpha = 10^{-3}$, and solve the denoising problem for $\beta = 10, 1, 0.1, 0.01, 0.001, 0.0001$. How do the number of Newton iterations and the cumulative number of conjugate gradient iterations behave for decreasing β?

c) Set $\alpha = 10^{-3}$ and $\beta = 10^{-2}$ and solve the denoising problem for different resolutions. In cell 3 set the number of pixels (`nx` , `ny`) to $(128, 128)$, $(256, 256)$, and $(512, 512)$ and report the number of Newton iterations and the cumulative number of conjugate gradient iterations. What do you observe?