Homework 5: Optimization and Differential Equations

Due December 7, 2023 (100 points)

Problem 1 (25 points)**.** In this problem, we develop the well-known *iterative shrinkage-thresholding algorithm* (ISTA), which has seen renewed interest thanks to its machine learning applications.

(a) Show that the iteration from gradient descent $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$ can be rewritten in *proximal form* as

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \left[f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2\alpha} \|\mathbf{x} - \mathbf{x}_k\|_2^2 \right].$$

(b) Suppose we wish to minimize a sum $f(\mathbf{x}) + g(\mathbf{x})$. Based on the previous part, ISTA attempts to combine exact optimization for g with gradient descent on f:

$$\mathbf{x}_{k+1} \equiv \operatorname*{argmin}_{\mathbf{x}} \left[f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2\alpha} \|\mathbf{x} - \mathbf{x}_k\|_2^2 + g(\mathbf{x}) \right].$$

Derive the alternative form

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{x}} \left[g(\mathbf{x}) + \frac{1}{2\alpha} \|\mathbf{x} - (\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))\|_2^2 \right].$$

(c) Derive a formula for ISTA iterations when $g(\mathbf{x}) = \lambda \|\mathbf{x}\|_1$, where $\lambda > 0$. You can write your answer using linear algebra operations and evaluations of $\nabla f(\cdot)$.

Problem 2 (25 points)**.** The swing angle θ of a pendulum under gravity satisfies the ODE: $\theta'' = -\sin\theta$, where $|\theta(0)| < \pi$ and $\theta'(0) = 0$.

(a) Suppose $\theta(t)$ satisfies the ODE. Show that the following value (representing the energy of the system) is constant as a function of t:

$$E(t) := \frac{1}{2}(\theta')^2 - \cos\theta.$$

(b) Many ODE integrators drift away from the desired output as time progresses over larger periods. For instance, forward Euler can add energy to a system by overshooting, while backward Euler tends to damp out motion and remove energy. In many computer graphics applications, quality long-term behavior can be prioritized, since large-scale issues cause visual artifacts. The class of *symplectic* integrators is designed to avoid this issue.

Denote $\omega \equiv \theta'$. The *symplectic Euler* scheme makes a series of estimates $\theta_0, \theta_1, \theta_2, \theta_3, \ldots$ and $\omega_0, \omega_1, \omega_2, \omega_3, \ldots$ at time $t = 0, h, 2h, 3h, \ldots$ using the following iteration:

$$\begin{aligned} \theta_{k+1} &= \theta_k + h\omega_k \\ \omega_{k+1} &= \omega_k - h \sin\theta_{k+1}. \end{aligned}$$

Define

$$E_k := \frac{1}{2}\omega_k^2 - \cos\theta_k.$$

Show that $E_{k+1} = E_k + O(h^2)$.

(c) Suppose we make the small-angle approximation $\sin\theta \approx \theta$ and decide to solve the linear ODE $\theta'' = -\theta$ instead. Now, symplectic Euler takes the following form:

$$\begin{aligned} \theta_{k+1} &= \theta_k + h\omega_k \\ \omega_{k+1} &= \omega_k - h\theta_{k+1}. \end{aligned}$$

Write a 2×2 matrix A such that

$$\begin{pmatrix} \theta_{k+1} \\ \omega_{k+1} \end{pmatrix} = A \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix}.$$

(d) If we define $E_k := \omega_k^2 + h\omega_k\theta_k + \theta_k^2$, show that $E_{k+1} = E_k$ in the iteration from (c). In other words, E_k is constant from time step to time step.

Problem 3 (50 points)**.** *Graph-based semi-supervised learning* (SSL) algorithms predict a quantity or label associated with the nodes of a graph given labels on a few of its vertices. For instance, under the (dubious) assumption that friends are likely to have similar incomes, it could be used to predict the annual incomes of all members of a social network given the incomes of a few of its members.

(a) Take $G = (V, E)$ to be a connected graph, and define $f_0 : V_0 \to \mathbb{R}$ to be a set of scalar-valued labels associated with the nodes of a subset $V_0 \subseteq V$. The *Dirichlet energy* of a full assignment of labels $f : V \to \mathbb{R}$ is given by

$$E[f] := \sum_{(v_1,v_2)\in E} (f(v_2) - f(v_1))^2.$$

Show how $E[f]$ can be minimized over f satisfying $f(v_0) = f_0(v_0)$ for all $v_0 \in V_0$ using a linear solve.

(b) Suppose f is the result of the optimization from Exercise (a). Prove the *discrete maximum principle*:

$$\max_{v\in V} f(v) = \max_{v_0\in V_0} f_0(v_0).$$

Relate this result to a physical interpretation of Laplace's equation.

(c) Fill in the code for a conjugate gradient solver `conjugate_gradient(A,b)` that takes positive definite $A \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ as input and outputs a solution to $A\mathbf{x} = \mathbf{b}$ using the conjugate gradients method.

(d) Fill in the code to compute the *k-nearest neighbor graph* of a set of n points $\mathbf{p}$. This is an undirected graph with one vertex $v_\mathbf{p}$ per point $\mathbf{p}$, and an edge $(v_\mathbf{p}, v_\mathbf{q})$ if and only if $\mathbf{p}$ is among the k-nearest neighbors of $\mathbf{q}$ or $\mathbf{q}$ is among the k-nearest neighbors of $\mathbf{p}$ (with respect to the Euclidean distance $\|\mathbf{p} - \mathbf{q}\|_2$).

(e) You have been provided a set of 1000 points $\mathbf{x}_i \in \mathbb{R}^2$, along with a dictionary storing the indices of two points $\mathbf{p}_0, \mathbf{q}_0$ in V_0 and their corresponding labels $f(\mathbf{p}_0), f(\mathbf{q}_0)$. Construct the k-nearest neighbor graph of these points for $k \in \{5, 10, 100\}$, and for each graph, use your conjugate gradient code to solve the linear system you derived in the first part of this question for a vector of predicted labels $f_k \in \mathbb{R}^{1000}$. Plot the k-nearest neighbor graph for each k. Plot the value of f_k at each point $\mathbf{x}_i$ and for each k. What happens when k is too small? What happens when k is too large?
Hint: Make sure the matrix you use in conjugate gradients is symmetric and positive definite.

Problem 4 (Extra credit)**.** Write up a clean solution to any midterm problems for which you did not receive full credit. We will offer you 50% of the points back if your new answers are correct. You may do this revision for as many midterm problems as you like.