CS4220 Assignment 7   Due: 4/28 (Monday) at 11pm

You must work either on your own or with one partner. You may discuss background issues and general solution strategies with others, but the solutions you submit must be the work of just you (and your partner). If you work with a partner, you and your partner must first register as a group in CMS and then submit your work as a group. One point may be deducted for poor style.

Topics: Sparse eigenvalue problems, `eigs`, nonlinear systems, `fsolve`, steepest descent with line search.

1 An Inverse Eigenvalue Problem

Given $\lambda_1 < \lambda_2 < \cdots < \lambda_n$, we wish to determine $d_1, \ldots, d_n$ so that the eigenvalues of the symmetric tridiagonal matrix

$$T = \begin{bmatrix} d_1 & 0.1 & 0 & \cdots & 0 \\ 0.1 & d_2 & 0.1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & 0.1 & d_{n-1} & 0.1 \\ 0 & \cdots & 0 & 0.1 & d_n \end{bmatrix}$$

are $\lambda_1, \ldots, \lambda_n$. Write a function `d = InvEig(lambda)` that does this. Here, `lambda` is a column n-vector whose components are the required eigenvalues arranged from most negative to most positive. You may assume that the separation between any two eigenvalues is at least 1, an assumption that makes the following theorem of interest:

If the Gershgorin disks of a matrix are disjoint, then each disk houses exactly one eigenvalue

Your implementation of `InvEig` should make effective use of `fsolve` (default parameters fine) and `eig`. Submit `InvEig` to CMS. This problem is worth 5 points.

2 Eigenvalues of a Data Sparse Symmetric Matrix

We wish to compute the k most positive eigenvalues and the corresponding eigenvectors of the matrix $A + UB^{-1}U^T$ where (a) $A \in \mathbb{R}^{n\times n}$ is symmetric and in sparse format, (b) $B \in \mathbb{R}^{r\times r}$ is symmetric and positive definite, and dense, and (c) $U \in \mathbb{R}^{n\times r}$ is in sparse format. Assume that $n >> r$. Write a function

`[Q,d] = DataSparseEig(A,B,U,k)`

that does this. In particular, `d` should be a column k-vector that houses the k largest eigenvalues and `Q` should be an n-by-r matrix with the property that `Q(:,j)` houses a unit eigenvector associated with `d(j)`. Make effective use of `eigs`. Submit `DataSparseEig` to CMS. This problem is worth 5 points.

3 Nearest Point on an Ellipsoid

An ellipsoid $\mathcal{E}$ in $\mathbb{R}^3$ is characterized by a center $r \in \mathbb{R}^3$, three semiaxes a, b, and c, and an orthogonal "rotation matrix" $Q \in \mathbb{R}^{3\times 3}$. In particular,

$$\mathcal{E} = \left\{ x \in \mathbb{R}^3 \mid x = r + Q \begin{bmatrix} a\cos(\phi)\cos(\theta) \\ b\sin(\phi)\cos(\theta) \\ c\sin(\theta) \end{bmatrix},\ -\pi \le \phi \le \pi,\ -\pi/2 \le \theta \le \pi/2 \right\}$$

Given a vector $p \in \mathbb{R}^3$ and an ellipsoid $\mathcal{E}$, we wish to determine that point $x_* \in \mathcal{E}$ that is closest to p. Write a function `xStar = Closest2E(E,p,tol,itMax)` that does this. Implementation details:

- The input parameter `E` should be a structure with `E.r` housing the center of the ellipsoid, `E.a`, `E.b`, and `E.c` housing the semiaxes of the ellipsoid, and `E.Q` housing the orthogonal rotation matrix associated with the ellipsoid.
- You may assume that p is outside the ellipsoid.
- Your function should use the method of steepest descent (see below) to minimize the function.

$$f(\begin{bmatrix} \phi \\ \theta \end{bmatrix}) = \left\| \left(r + Q \begin{bmatrix} a\cos(\phi)\cos(\theta) \\ b\sin(\phi)\cos(\theta) \\ c\sin(\theta) \end{bmatrix} \right) - p \right\|_2^2$$

- Supply comments that explain your choice for a starting value. Arrange and comment your code in such a way that we can easily check your f-evaluations, gradient evaluations, and step length computations.

Submit `Closest2E` to CMS. This problem is worth 10 points.

Here is the steepest descent framework that your method should implement (think of x as $\begin{bmatrix} \phi \\ \theta \end{bmatrix}$):

> Determine an initial guess x_c.
>
> Compute $f_c = f(x_c)$ and $g_c = \nabla f(x_c)$
>
> $its = 0$
>
> **while** $\| g_c \| \geq tol$ and $its < itMax$
>
> Let $\tilde{f}_c(t) = f(x_c - t \cdot g_c)$ and let q_c be a quadratic function that satisfies
>
> $q_c(0) = \tilde{f}(0)$, $q_c(1) = \tilde{f}(1)$, and $q_c'(0) = \tilde{f}'(0)$.
>
> Let t_c minimize $q_c(t)$ over the interval $[0, 1]$
>
> $x_c = x_c - t_c g_c$
>
> $g_c = \nabla f(x_c)$
>
> $its = its + 1$
>
> **end**
>
> $x_* = x_c$

Take care to avoid redundant f-evaluations and gradient evaluations.