Midterm
Due: Monday, Oct 28.

Choose any five problems. You may use the textbook or other references (with citation). You may discuss problems with me, but not with each other.

Test your codes, and document your tests. There will be no credit for code that is clearly broken and untested. There will also be points deducted if you use inv or equivalent constructs like A\eye(size(A)).

Bonus point (not one of the five questions): Tell me something you particularly like about the class or give me a suggestion to improve it.

1. Consider the function \( f(x) = \sqrt{1 + x} - \sqrt{1 - x} \) defined on \([-1, 1]\).

   Give an example that illustrates that the following naive MATLAB algorithm can have poor relative accuracy in floating point:

   \[
   \text{function } f = \text{mtbadf}(x) \\
   f = \sqrt{1+x} - \sqrt{1-x};
   \]

   Write an alternate formulation mtgoodf that has good relative accuracy (relative error of a few machine epsilon) over the entire range \([-1, 1]\).

2. Suppose \( PM = LU \) for a square matrix \( M \in \mathbb{R}^{n \times n} \), and let \( f(s) \) be the \((1,1)\) entry for \((M + sE)^{-1}\), where \( E \in \mathbb{R}^{n \times n} \) is a given fixed matrix. Following routine to compute \( f'(0) \). Your code should take \( O(n^2) \) time.

   \[
   \text{function } [dfds] = \text{mtderiv}(L,U,P,E)
   \]

3. Suppose \( A \) is symmetric positive definite and the Cholesky factorization \( A = R^T R \) is given. Write a function to compute an \( LU \) factorization of \( A \) in \( O(n^2) \) time.

   \[
   \text{function } [L,U] = \text{mtcholtolu}(R)
   \]

4. Suppose \( \hat{A} \in \mathbb{R}^{n \times n} \), and let \( C \in \mathbb{R}^{n \times k} \) with \( k < n \). Assuming that \( C \) is not rank deficient, find \( B \in \mathbb{R}^{k \times n} \) to minimize \( \|A - CB\|_F \).

   \[
   \text{function } [B] = \text{mtfindB}(A,C)
   \]
5. Let \((x_i, y_i)\) for \(i = 1, \ldots, n\) be the coordinates of \(n\) points in the plane. Let \(D \in \mathbb{R}^{n \times n}\) be a matrix of squared Euclidean distances, i.e. \(d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2\). Supposing the first three points are not co-linear, write a function to recover all the points from the distance matrix and the coordinates of these two points.

\[
\text{function } [x,y] = \text{mtrecover}(D, x1, y1, x2, y2, x3, y3)\]

6. Suppose \(L\) is nonsingular and lower bidiagonal (i.e. \(l_{ij}\) is only nonzero for \(i = j\) or \(i = j + 1\)). Write a code to compute \(\kappa_\infty(L)\) in \(O(n)\) time.

\[
\text{function } [\text{condL}] = \text{mtcondL}(L)\]

*Note:* The MATLAB `cond` function is not \(O(n)\)!