HW for 2020-02-07
(due: 2020-02-14)

You may (and should) talk about problems with each other and with me, providing attribution for any good ideas you might get. Your final write-up should be your own.

1: Growing a system Suppose \( PA = LU \) is given. Write an \( O(n^2) \) time algorithm to extend the factorization to an LU factorization of \( M = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \).

Write your code as a function that takes \( P, L, U, b, c, \) and \( d \) as inputs, and returns extended matrices \( \hat{P}, \hat{L}, \) and \( \hat{U} \). You should verify that your code is correct for random choices of \( A, b, c, \) and \( d \) by checking the backward error, i.e. \( \| \hat{P} M - \hat{L} \hat{U} \|_F / \| M \|_F \) should be small.

2: Shrinking back Suppose \( PA = LU \) is given. Consider the system \( Ax = b + re_i \) where \( x_j = 0 \); that is, we allow the \( i \)th equation not to be satisfied (by an unknown amount \( r \)), but enforce that \( x_j = 0 \). Express this new problem in terms of a bordered system, and use block elimination to give an \( O(n^2) \) code to compute \( x \) and \( r \).

3: Iterative refinement and bordering The straightforward algorithm in problem 2 runs into stability problems when \( A \) is nearly rank deficient, even if the modified system is well-conditioned. Illustrate the problem with the matrix

\[
A = \begin{bmatrix} 1 & 1 \\ 1 & 1 + \delta \end{bmatrix}
\]

for \( \delta = 10^{-12} \). Show experimentally that a step of iterative refinement fixes the issue.