

CS 6210: HOMEWORK 3

Instructor: Anil Damle

Due: March 8, 2022

POLICIES

You may discuss the homework problems freely with other students, but please refrain from looking at their code or writeups (or sharing your own). Ultimately, you must implement your own code and write up your own solution to be turned in. Your solution, including plots and requested output from your code should be typeset and submitted via the Gradescope as a pdf file. Additionally, please submit any code written for the assignment. This can be done by either including it in your solution as an appendix, or uploading it as a zip file to the separate Gradescope assignment.

QUESTION 1:

For square non-singular A implement the computation of a QR factorization via Householder triangularization. This now gives a means for solving linear systems $Ax = b$. For the following you may use your existing code for computing LU decompositions with partial pivoting. (If you did not complete the previous homework or did not end up with working code you can use a built-in implementation.)

- (a) Compare and contrast the runtimes of your QR factorization and your LU factorization with partial pivoting. As part of this, show your implementation of Householder QR is $\mathcal{O}(n^3)$.
- (b) Build a set of problems that are increasingly ill-conditioned. (These should be general dense matrices and not something explicitly structured, i.e., do not use diagonal matrices.) Using this set of matrices explore the relationship between the condition number of the problem and your ability to accurately solve $Ax = b$ using your QR factorization routine. How does this compare to what you observe when using LU with partial pivoting? What about for some of the “worst case” problems we considered on the last homework?

QUESTION 2 (UNGRADED BUT FUN, AND MAYBE A GOOD CONCEPTUAL WARM UP FOR LATER IN THE SEMESTER):

Say we would like to compute a QR factorization of A^k for some real, square A and (potentially large) integer k .

- (a) Even if it is feasible to simply construct A^k , is it a good idea numerically to compute its QR factorization directly? Justify your response.
- (b) Develop a scheme to compute a QR factorization of A^k via a sequence of k QR factorizations of matrices whose condition number is never larger than that of A itself. As long as you adhere to this constraint, for the purposes of this problem you may assume that computing matrix matrix products with the resulting Q and/or R matrices is fine numerically.