18.335 Take-Home Midterm Exam: Spring 2020

Posted 4:30pm Friday April 10, due **4:30pm Saturday April 11.**

Problem 0: Honor code

Copy and sign the following in your solutions:

I have not used any resources to complete this exam other than my own 18.335 notes, the textbook, running my own Julia code, and posted 18.335 course materials.

____

your signature

Problem 1: (20+5 points)

For an ordinary least-squares problem with an $m \times n$ matrix A of rank $n \leq m$, that is computing the function $f(b) = \hat{x} =$ minimizer of $\|b - Ax\|_2^2$, we showed via $A = \hat{Q}\hat{R}$ factorization in class that the condition number (in the L_2 norm) of $f(b) = \hat{R}^{-1}\hat{Q}^* b$ is bounded above by $\kappa(\hat{R}) = \kappa(A) = \sigma_1/\sigma_n$ where the σ_k are the singular values of A. [Recall that the condition number of $f(b) = Bb$ is bounded above by $\kappa(B)$; this is a "tight" bound that is hit when b is a singular vector of B.]

(a) Now, instead, consider the "regularized" least-square problem

$$g(b) = \hat{x} = \text{minimizer of } \|b - Ax\|_2^2 + \alpha \|x\|_2^2,$$

where $\alpha > 0$ is some constant. **Give a similar (tight) upper bound on the condition number** of $g(b)$ in the L_2 norm, in terms α and the singular values of A? You should find that a **larger α gives a smaller condition number**. (*Hint:* try to write g in terms of an ordinary least-square problem for a new matrix.)

(b) If A is a badly conditioned matrix, a larger α therefore trades off ______________ for ________________ in trying to find a "best-fit" $\hat{x}$.

Problem 2: (15+5+5 points)

Let $f(x) = \sqrt{1+x^2} - 1$, and consider its computation in floating-point arithmetic.

(a) Explain the difficulty of naively computing $f(x)$ accurately for a small value of $|x|$ and show how the difficulty can be circumvented.

To get full marks, do *not* use an infinite series like a Taylor expansion—your method to compute f should ideally involve a fixed number of $\{\sqrt{}, \pm, \times, \div\}$ operations regardless of the precision.

(b) Show that $f(x)$ is *not* ill-conditioned for small $|x|$.

(c) Reconcile your answers from (a) and (b): if f is well-conditioned, how can there be a difficulty in computing it? (Don't write a long essay, just a few words.)

Problem 3: (10+10+5 points)

(a) Let A be a 6×6 real-symmetric matrix with eigenvalues $\{3, -3, 2, -2, 1, -1\}$. Suppose we perform the usual unshifted QR iteration [$A^{(0)} = A$, $A^{(k)} = R^{(k-1)}Q^{(k-1)}$ where $Q^{(k-1)}R^{(k-1)}$ is the QR factorization of $A^{(k-1)}$]. What is the **nonzero pattern** of the matrix $A^{(k)}$ for a large value k? **Why**?

(b) Suppose that unshifted QR iteration on a Hermitian matrix A produces $A^{(k)}$ after k iterations. If we compute the eigenvalues of any $d \times d$ block D along the diagonal of $A^{(k)}$, explain why the eigenvalues of D are a form of *Ritz value* of A corresponding to a Rayleigh-Ritz procedure for a certain subspace.

(c) Explain how you could use the answer from part (b) to accurately compute all the eigenvalues of A from part (a) using a small (finite) number of $\{\sqrt{}, \pm, \times, \div\}$ operations, given $A^{(k)}$ for a large k.

Problem 4: (10+15 points)

Suppose that we compute the transpose of a square matrix A in-place using obvious algorithm

```
function my_transpose!(A::Matrix)
    m, n = size(A)
    m == n || error("my_transpose! requires a square matrix")
    for i = 1:m
        for j = i+1:m
            A[i,j], A[j,i] = A[j,i], A[i,j] # swap ij and ji entries
        end
    end
end
```

(a) If we have an ideal cache of size Z, with cache-line length L, give the asymptotic (large m) cache complexity ("big O", i.e. ignoring constant factors) as a function of m, Z, L. (Loading a *whole cache line* into cache $= 1$ miss. As discussed in class, a Julia matrix A is stored in column-major order, i.e. with contiguous columns in memory.)

(b) Describe a modified algorithm with better cache performance, and give its asymptotic cache complexity.