18.335 Take-Home Midterm Exam: Spring 2023

Posted Friday 12:30pm April 14, due **11:59pm Monday April 17.**

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: (32 points)

Given two real vectors $u = (u_1, u_2, \ldots, u_n)^T$ and $v = (v_1, v_2, \ldots, v_n)^T$, computing the dot product $f(u,v) = u_1v_1 + u_2v_2 + \cdots + u_nv_n = u^Tv$ in floating point arithmetic with left to right summation is backward stable. The computed dot product $\hat{f}(u,v)$ satisfies the *component-wise* backward error criteria

$$\hat{f}(u,v) = (u+\delta u)^T v, \qquad \text{where} \qquad |\delta u| \le n\varepsilon_{\text{mach}}|u| + \mathscr{O}(\varepsilon_{\text{mach}}^2).$$

The notation $|w|$ indicates the vector $|w| = (|w_1|, |w_2|, \ldots, |w_n|)^T$, i.e., the vector obtained by taking the absolute value of each entry of w.

(a) Using the dot product algorithm $\hat{f}(u,v)$, derive an algorithm $\hat{g}(A,b)$ for computing the matrix-vector product $g(A,b) = Ab$ in floating point arithmetic, and show that it satisfies the component-wise backward stability criteria

$$\hat{g}(A,b) = (A+\delta A)b, \qquad \text{where} \qquad |\delta A| \le n\varepsilon_{\text{mach}}|A| + \mathscr{O}(\varepsilon_{\text{mach}}^2),$$

where the notation $|B|$ indicates the matrix obtained by taking the absolute value of each entry of B.

(b) Suppose the algorithm $\hat{g}(A,b)$ is used to compute matrix-matrix products $C = AB$ by computing one column of the matrix C at a time. Is the resulting floating-point algorithm $\hat{h}(A,B)$ component-wise backward stable in the sense that there is a matrix δA such that

$$\hat{h}(A,B) = (A+\delta A)B, \qquad \text{where} \qquad |\delta A| \le n\varepsilon_{\text{mach}}|A| + \mathscr{O}(\varepsilon_{\text{mach}}^2)?$$

Explain why or why not.

Problem 2: (32 points)

Given an n-dimensional subspace $\mathscr{V}$, the standard Rayleigh–Ritz projection approximates a few ($n \ll m$) eigenvalues of an $m \times m$ matrix A by finding a scalar λ and $x \in \mathscr{V}$ such that $Ax - \lambda x \perp \mathscr{V}$, i.e., the residual is perpendicular to the subspace. A *two-sided* Rayleigh–Ritz projection uses a second subspace $\mathscr{W}$ (not orthogonal to $\mathscr{V}$) and searches for a scalar λ and $x \in \mathscr{V}$ such that

$$Ax - \lambda x \perp \mathscr{W}, \qquad \text{and} \qquad x \in \mathscr{V}, \tag{1}$$

i.e., the residual is perpendicular to the *second* subspace.

(a) Let V and W be orthonormal bases for $\mathscr{V}$ and $\mathscr{W}$, and let λ and w solve the eigenvalue problem $Bw = \lambda Mw$, where $B = W^TAV$ and $M = W^TV$. Show that λ and $x = Vw$ satisfy the criteria in (1).

(b) Show that if $\mathscr{V}$ is a *right invariant subspace* of A with associated *left invariant subspace* $\mathscr{W}$, then λ and x in part (a) satisfy the original eigenvalue problem $Ax = \lambda x$. HINT: show that $\|W^T(Ax - \lambda x)\| = 0$ can only happen if $Ax = \lambda x$.

A *right invariant subspace* of A is invariant under multiplication by A from the right: $Ax \in \mathscr{V}$ for all $x \in \mathscr{V}$. A *left invariant subspace* of A is invariant under multiplication by A from the left: $A^T y \in \mathscr{W}$ for all $y \in \mathscr{W}$.

Problem 2: (36 points)

The method of Generalized Minimal RESiduals (GMRES) uses n iterations of the Arnoldi method to construct a sequence of approximate solutions $x_1, x_2, \ldots, x_n$ to the $m \times m$ linear system $Ax = b$. At the n^{th} iteration, the approximate solution $x_n = Q_n y_n$ is constructed by solving the least-squares problem,

$$y_n = \operatorname{argmin}_y \|\tilde{H}_n y - \|b\| e_1\|,$$

where $\tilde{H}_n$ is an $(n+1) \times n$ upper Hessenberg matrix and Q_n is the usual orthonormal basis for the Krylov subspace $\mathscr{K}_n(A, b) = \text{span}\{b, Ab, A^2 b, \ldots, A^{n-1} b\}$.

(a) Describe an algorithm based on Givens rotations that exploits the upper Hessenberg structure of $\tilde{H}_n$ to solve the $(n+1) \times n$ least-squares problem in $\mathscr{O}(n^2)$ flops.

(b) If the QR factorization $\tilde{H}_{n-1} = \Omega_{n-1} R_{n-1}$ is known from the previous iteration, explain how to update the QR factorization to $\tilde{H}_n = \Omega_n R_n$ cheaply using a single Givens rotation.

(c) Using your result from part (b), explain how the solution to the least-squares problem can also be updated cheaply from the solution at the previous iteration.

(d) What is the approximate flop count for updating the least-squares solution at the n^{th} step of GMRES? You may use big-O notation to express the asymptotic scaling in n.