End Semester Lab Assessment

MA423: Matrix Computations 13 November, 2025 S. Bora

Total Marks: 40

Important instructions:

(i) **Switch to `format long e` for all experiments.**

(ii) **Submit a <u>single livescript file</u> that contains all comments, answers and codes necessary to produce the required outputs. Ensure that the answers are correctly numbered and the file does not include any irrelevant material. The filename of the livescript file should be <u>MA423ESYourrollnumber.mlx</u>**

1. Let A be a *properly upper Hessenberg real* matrix of size $n \times n$ and $p(A) = (A - \rho I)(A - \mu I)$ where ρ and μ are eigenvalues of $A(n-1:n, n-1:n)$. The only non-zero entries of the first column of $p(A)$ are its first three entries and they form the vector

$$x = \begin{bmatrix} (a_{11}-\rho)(a_{11}-\mu) + a_{12}a_{21} \\ a_{21}(a_{11}+a_{22}-\rho-\mu) \\ a_{32}a_{21} \end{bmatrix}.$$

   (a) Compute x from the entries of A so that no complex arithemetic is used even if ρ and μ are not real numbers. **(2)**

   (b) Use the `reflect.m` code to compute u, γ, and τ such that $Qx = [-\tau\ 0\ 0]^T$ for $Q = I_3 - \gamma uu^T$. Illustrate the correctness of your output by using the `applreflect.m` program to compute Qx for two randomly generated properly upper Hessenberg matrices of size 10 and 12. **(9)**

   [**Marking Scheme:** `reflect.m` [4] + `applreflect.m` [3] + verification (for correct output only) [2]]

   (c) Make efficient use of `applreflect.m` to write `B = Bulge(A,u,gamma,tau)` to compute $B = \begin{bmatrix} Q & \\ & I_{n-3} \end{bmatrix} A \begin{bmatrix} Q & \\ & I_{n-3} \end{bmatrix}$ in $O(n)$ flops. **(5)**

   [**Marking Scheme:** `Bulge.m` [3] + verification (for correct output only) [2]]

2. Write a function program $[\mathtt{L}, \mathtt{U}, \mathtt{p}] = \mathtt{geppհess}(\mathtt{A})$ that performs GEPP on an $n \times n$ upper Hessenberg matrix A in $O(n^2)$ flops to produce an LU decomposition of $A(p,:)$, p being the permutation vector that records the row interchanges. Check the correctness of your output on one randomly generated upper Hessenberg matrix of size 10. **(10)**

   [**Marking Scheme:** `gepphess.m` [8] + verification (for correct outputs only) [2]]

3. Given a real $n \times n$ matrix A, write a function program $[\mathtt{iter}, \mathtt{lambda}] = \mathtt{Rayleigh}(\mathtt{A}, \mathtt{x}, \mathtt{k})$ that *efficiently* performs `k` Shift and Invert iterations with Rayleigh quotient shifts on $\mathtt{A} \in \mathbb{C}^{\mathtt{n}\times\mathtt{n}}$ with initial vector $\mathtt{x} \in \mathbb{C}^{\mathtt{n}}$ such that the cost per iteration is $O(n^2)$ flops. The output `iter` should be an $n \times k$ matrix whose jth column is the jth iterate q_j and `lambda` should be an eigenvalue of A if the iterations are converging. **(5)**

4. Run `[iter, lambda] = Rayleigh(A, x, k)` on the following matrices with $x = [1\ 1\ 1]^T$ and $k = 15$.

$$(i)\ A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 9 & 2 \\ 0 & -1 & 2 \end{bmatrix} \qquad (ii)\ A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 9 & 2 \\ -4 & -1 & 2 \end{bmatrix} \qquad (iii)\ A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 3 & 2 \\ -4 & -1 & 2 \end{bmatrix}$$

Do you observe quadratic convergence in any of them? Justify your answer. **(9)**