MATH 8610 (SPRING 2023) HOMEWORK 7

Assigned 04/02/23, due 04/10/23 (Monday) by 11:59pm.

Instructor: Dr. Fei Xue, Martin O-203, fxue@clemson.edu.

1. **[Q1]** (15 pts) **(a)** Let H be the initial input matrix for the shifted QR iteration. Show that $(H-\mu^{(k)}I)\cdots(H-\mu^{(2)}I)(H-\mu^{(1)}I) = \underline{Q}^{(k)}\underline{R}^{(k)}$ (in practice, H is upper Hessenberg, but we do not need this assumption here)

   (b) Other than the Wilkinson shift, we may also let $\mu^{(k)} = h_{nn}^{(k-1)}$ if $h_{n(n-1)}^{(k-1)}$ is small. Assume, for example, that $H^{(k-1)} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \delta & h_{nn}^{(k-1)} \end{bmatrix}$. After the application of $n-2$ Givens rotations to $H^{(k-1)} - h_{nn}^{(k-1)}I$, we have the intermediate matrix $H_{tmp}^{(k-1)} = \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & \delta & 0 \end{bmatrix}$ (make sure you understand why it is of this form), and the last Givens rotation is needed on the left before we compute $H^{(k)}$ by transposed Givens rotations. Show that the new matrix $H^{(k)} = R^{(k)}Q^{(k)} + h_{nn}^{(k-1)}I$ satisfies $h_{n(n-1)}^{(k)} = -\frac{b\delta^2}{a^2+\delta^2}$. What does this observation suggest, if $\left|h_{n(n-1)}^{(k-1)}\right| = |\delta| \ll 1$, and either $|b| < 2|a|$ (δ can be arbitrary) or if $|\delta| < \frac{a^2}{|b|}$?

   (c) What can we say about $h_{n(n-1)}^{(k)}$ if A is real symmetric, such that $H^{(k-1)}$ is also real symmetric (hence tridiagonal)? In particular, does this entry decrease more slowly or more rapidly in the symmetric case than in the nonsymmetric case?

2. **[Q2]** (10 pts) Implement the single-shift QR step in MATLAB; that is, given an upper Hessenberg $H^{(k-1)}$ and shift $\mu^{(k)}$, we compute $Q^{(k)}R^{(k)} = H^{(k-1)} - \mu^{(k)}I$ by Givens rotations and then use these Givens rotations to compute $H^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I$. Make simple changes in my code to enforce the use of single (Wilkinson) shift only, even if complex arithmetic is needed. Assemble your single shift code with the uploaded subroutines. Test it with the matrix obtained by

   ```
   load west0479;
   A = full(west0479);
   ```

   Compare the eigenvalues of your final $H^{(k)}$ (use `ordeig`) with those of A. Be aware that the ordering of eigenvalues must be consistent to make a meaningful comparison.

3. **[Q3]** (10 pts) Implement the Arnoldi's method without and with reorthogonalization, and test the orthogonality of the column vectors in U_{50} for the matrix A generated by `u = cos((0:2048)/2048*pi); A = vander(u);` Is the reorthogonalization effective for generating an orthonormal basis?

   Use Arnoldi with reorthogonalization to compute the 11 dominant eigenvalues and eigenvectors of `aerofoil_new`, using $m = 30, 60, 100$, and 150 dimensional Krylov subspaces. For each m, plot all eigenvalues $\{\lambda_i\}_{i=1}^n$ of A together with the eigenvalues $\{\mu_i\}_{i=1}^m$ of H_m on the complex plane. Intuitively, how do $\{\mu_i\}_{i=1}^m$ approximate $\{\lambda_i\}_{i=1}^n$ as m increases? Give the relative eigenresidual norm $\frac{\|AU_m w_i - \mu_i U_m w_i\|_2}{\|AU_m w_i\|_2}$ $(1 \le i \le 11)$ of the desired eigenpairs for each m in a table.

Algorithm 1 Arnoldi's method for computing p dominant eigenvalues of A

Input: Matrix $A \in \mathbb{R}^{n\times n}$, $u_1 \in \mathbb{R}^n$ with $\|u_1\|_2 = 1$ (typically set as a random vector, normalized), maximum number of Arnoldi step m, and a tolerance $\delta > 0$

Output: Eigenpairs $\{(\lambda_i, v_i)\}_{i=1}^p$, where $\{\lambda_i\}$ are the dominant eigenvalues of A

Initialize $U_{m+1} = [u_1, u_2, \ldots, u_{m+1}] = [u_1,\ 0_{n\times m}]$, $H = [h_{ij}] = 0_{(m+1)\times m}$;
for $k = 1, 2, \ldots, m$ **do**
  $w = Au_k$;
  for $j = 1, 2, \ldots, k$ **do**
    $h_{jk} \leftarrow u_j^T w$; $w \leftarrow w - u_j h_{jk}$;
  end for
  for $j = 1, 2, \ldots, k$ **do**
    $\Delta h \leftarrow u_j^T w$; $h_{jk} \leftarrow h_{jk} + \Delta h$; $w \leftarrow w - u_j \Delta h$; (optional reorthogonalization)
  end for
  $h_{(k+1)k} \leftarrow \|w\|_2$;
  $u_{k+1} \leftarrow w/h_{(k+1)k}$;
end for
Compute the p dominant eigenpairs $\{(\mu_i, w_i)\}$ of the leading $m \times m$ principal of H.
Then $\{(\mu_i, U_m w_i)\} \approx \{(\lambda_i, v_i)\}$ are the desired dominant eigenpair approximations.

4. **[Q4***] (5 extra pts) Read the implicit double-shifted QR step for real nonsymmetric matrices and the overall QR iteration. Then read my codes to see how the described algorithms are implemented. Debug my code to compare numerically if the double shifted QR step gives a new upper Hessenberg matrix that is numerically the same as the upper Hessenberg matrix obtained by using the pair of complex conjugate shifts successively in two single-shifted QR steps. Use your own words to summarize (not to repeat) the overall QR iteration.