Homework problems that will be graded (P1 - P6, 30pts in total):

**P1.** (This is similar to exercise 5.3.33 from the textbook)

Let $H$ be an upper Hessenberg matrix. Show that the flop count for computing the $QR$ decomposition of $H$ is $O(n^2)$, assuming that the factor $Q$ is not assembled but left as a product of rotators.

**P2.** (This is similar to exercise 5.4.29 from the textbook)

Suppose $B = S^{-1}AS$ and $B + \delta B = S^{-1}(A + \delta A)S$. Show the following inequalities:

a) $\frac{1}{\kappa(S)} \|A\| \leq \|B\| \leq \kappa(S)\|A\|,$

b) $\frac{1}{\kappa(S)^2} \frac{\|\delta A\|}{\|A\|} \leq \frac{\|\delta B\|}{\|B\|} \leq \kappa(S)^2 \frac{\|\delta A\|}{\|A\|}.$

Here $\|\cdot\|$ denotes any matrix norm and $\kappa$ is the condition number associated with that norm.

**P3.** (This is similar to exercise 5.4.40 from the textbook)

A matrix $A \in \mathbb{C}^{n \times n}$ is called skew-Hermitian if $A^* = -A$.

Show that skew-Hermitian matrices have only purely imaginary eigenvalues, i.e., any eigenvalue of a skew-Hermitian matrix has the form $\lambda = ai$ with $a \in \mathbb{R}$.

*Hint:* Consider the Schur decomposition, and see what it means that $A^* = -A$.

**P4.** (This is similar to exercises 5.4.42 and 5.4.44 from the textbook)

Recall that a matrix $A \in \mathbb{C}^{n \times n}$ is called normal if $AA^* = A^*A$.

Prove that Hermitian, skew-Hermitian, and unitary matrices are normal.

**P5.** Show that if $A$ is a positive definite matrix, then all its eigenvalues are real and positive.

*One possibility:* Consider the Cholesky decomposition of the positive definite matrix and the SVD of its Cholesky factor.

**P6.** (MATLAB problem) Write a short code (or function) which implements the $QR$ iteration in MATLAB. You may use the MATLAB functions `hess` (which calculates the Hessenberg form of a matrix) and `qr` (calculating the $QR$ decomposition of a matrix).

For a matrix $A$, start your iteration with $A_1$ being the Hessenberg form of $A$, and calculate $A_{k+1}$ from $A_k$ via the given formulas.

To decide when to stop iterating, check that the 2-norm of the lower diagonal of $A_k$ is less than $10^{-4}$, and stop if it is; if not, keep going (it’s easiest to use a `while` loop).

For a matrix $X$, the lower diagonal can be obtained as $\text{diag}(X, -1)$. 

1
Keep track of how many iterations the code has taken using some counting variable that gets incremented each time a new $A_k$ is computed.

Your output should be

- the (approximate) eigenvalues of $A$, sorted in order of magnitude (this can be achieved using the Matlab routine `sort`);
- the number of iterations taken by the code to compute them.

Now that the code is written, complete the following two experiments and submit the results, along with the code.

a) Reset the random number generator by typing in `rng(112233)`; then run the code with input

$$A = \text{randn}(10) + i \times \text{randn}(10),$$

making sure the output is displayed.

b) Type in `clear all`; and then once again reset the random number generator by typing in `rng(112233)`; repeat part a), this time with the matrix $C$ obtained by typing

$$B = \text{randn}(6); \ C = B + B';$$