The same rules as for the first assignment apply. There will be 1 extra credit point for presentation, which will be given for carefully presented solutions, clean, descriptive plots and code.

1. **[Largest eigenvalues using the power method, 2+1pt]** The power method to compute the largest eigenvalue only requires the matrix-application rather than the matrix itself. To illustrate this, we use the power method to compute the largest eigenvalue of a matrix $A$, of which we only know its action on vectors.

   (a) Use the power method (with a reasonable stopping tolerance) to compute the largest eigenvalues of the symmetric matrix $A$, which is implicitly given through a function $A\text{fun()}$ that applies $A$ to vectors $v$. Use the convergence rate to estimate the magnitude of the next largest eigenvalue. Report results for $n = 10, 50$, where $v \in \mathbb{R}^{n^2}$ and $A \in \mathbb{R}^{n^2 \times n^2}$.

   (b) Compare your result for the largest two eigenvalues with what you obtain with an available eigenvalue solver.\(^1\)

2. **[Stability of eigenvalues for non-symmetric matrices, 2+1pt]** Let $A_\varepsilon$ be a family of matrices given by

   $$
   \begin{bmatrix}
   \lambda & 1 \\
   \varepsilon & \ddots & \ddots \\
   & \ddots & \ddots & 1 \\
   & & & \varepsilon & \lambda
   \end{bmatrix} \in \mathbb{R}^{n \times n}.
   $$

   Obviously, $A_0$ has $\lambda$ as its only eigenvalue with multiplicity $n$.

   (a) Show that for $\varepsilon > 0$, $A_\varepsilon$ has $n$ different eigenvalues given by

   $$
   \lambda_{\varepsilon,k} = \lambda + \varepsilon^{1/n} \exp(2\pi ik/n), \quad k = 0, \ldots, n - 1,
   $$

   and thus that $|\lambda - \lambda_{\varepsilon,k}| = \varepsilon^{1/n}$.

   (b) Based on the above result, what accuracy can be expected for the eigenvalues of $A_0$ when the machine epsilon is $10^{-16}$?

3. **[Finding all roots of a polynomial, 2+1pt]** An efficient way to find individual roots of a polynomial is to use Newton’s method. However, as we have seen, Newton’s method requires an initialization close to the root one wants to find, and it can be difficult to find all roots of a polynomial. Luckily, one can use the relation between eigenvalues and

---

\(^1\) Download the MATLAB function from [http://cims.nyu.edu/~stadler/num1/material/Afun.m](http://cims.nyu.edu/~stadler/num1/material/Afun.m). Inputs to that function should be vectors of size $n^2$, for $n \in \mathbb{N}$.

\(^2\) In MATLAB, `eigs` finds the largest eigenvalues of a matrix that is given through its application to vectors.
polynomial roots to find all roots of a given polynomial. Let us consider a polynomial of degree \( n \) with leading coefficient 1:

\[ p(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + x^n \quad \text{with} \ a_i \in \mathbb{R}. \]

(a) Show that \( p(x) \) is the characteristic polynomial of the matrix (sometimes called a companion matrix for \( p \))

\[
A_p := \begin{bmatrix}
0 & -a_0 \\
1 & -a_1 \\
& \ddots \\
& & 1 & -a_{n-1}
\end{bmatrix} \in \mathbb{R}^{n \times n}.
\]

Thus, the roots of \( p(x) \) can be computed as the eigenvalues of \( A_p \) using the QR algorithm (as implemented, e.g., in MATLAB’s \texttt{eig} function).

(b) Let us consider Wilkinson’s polynomial \( p_w(x) \) of order 15, i.e., a polynomial with the roots 1, 2, \ldots, 15:

\[
p_w(x) = (x - 1) \cdot (x - 2) \cdot \ldots \cdot (x - 15).
\]

The corresponding coefficients can be found using the \texttt{poly()} function. Use these coefficients in the matrix \( A_p \) to find the original roots again, and compute their error. Compare with the build-in method (called \texttt{roots()} for finding the roots of a polynomial. Is there a significant accuracy difference between your and the build-in method? \(^3\)

4. [Properties of the SVD, 1+2+1+1+1pt] Consider the singular value decomposition of \( A \in \mathbb{C}^{m \times n} \), i.e.,

\[
A = U \Sigma V^* \quad \text{(1)}
\]

where \( U \in \mathbb{C}^{m \times m} \) and \( V \in \mathbb{C}^{n \times n} \) are unitary, \( \Sigma \in \mathbb{R}^{m \times n} \) is diagonal and \( V^* \) denotes the adjoint of \( V \) (i.e., the complex conjugate matrix). The diagonal entries in \( \Sigma \) are real and nonnegative, and we assume them to be ordered in non-increasing order, i.e., \( \Sigma = \text{diag} (\sigma_1, \sigma_2, \ldots, \sigma_p) \), \( \sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq 0 \), where \( p = \min(m,n) \). If \( A \) is a real matrix, it has a real SVD, i.e., \( U, V \) can be chosen as real orthonormal matrices. The singular values are uniquely determined. If all the singular values are different, the columns of \( U \) and \( V \) (which are called the left and right singular vectors), are uniquely determined up to multiplication with scalars of absolute value 1.

(a) Show that the columns of \( U \) are the eigenvectors of \( AA^* \in \mathbb{C}^{m \times m} \), and that the columns of \( V \) are the eigenvectors of \( A^*A \in \mathbb{C}^{n \times n} \). What are the corresponding eigenvalues?

(b) Use the previous two properties to compute a (real) SVD for the matrix

\[
A = \begin{pmatrix}
1 & 0 & 3 \\
-3 & 0 & -1
\end{pmatrix}.
\]

\(^3\)For many MATLAB functions that do not use external libraries, you can see how they are implemented by typing \texttt{edit name.of.function}. Doing that for the \texttt{roots} function might help to answer this question.
(c) Let $m = n$ and $A$ be invertible. Using the SVD of $A$, give an expression for $A^{-1}$.

(d) Let $m < n$, and $\text{rank}(A) = m$, i.e., $A$ has full rank. Use the SVD of $A$ to compute the pseudoinverse $A^\dagger := A^* (AA^*)^{-1}$.

(e) The Frobenius norm\(^4\) of a matrix $A \in \mathbb{R}^{m \times n}$ is given by

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}.$$ 

Given the SVD of $A$, compute $\|A\|_F$.

5. **[SVD for image compression, 2pt extra credit]** Take your favorite black and white photo, consider the gray values of the pixels as a matrix, and compute the SVD of that matrix. Then, compress the matrix by only retaining the singular vectors corresponding to the largest few singular values (i.e., by zeroing out small singular values)\(^5\). What percentage of data of the original image data is necessary to obtain a reasonable image reconstruction? How do you think does this depend on the image? Discuss your observations.

---

\(^4\)Note that we have discussed matrix norms, that were induced by norms of vectors. The Frobenius norm is not naturally induced by a vector norm, since, for instance, an induced norm of the identity matrix should be 1.

\(^5\)Many examples of this can be found on the web, see for instance [http://math.arizona.edu/~brio/VIGRE/ThursdayTalk.pdf](http://math.arizona.edu/~brio/VIGRE/ThursdayTalk.pdf) or [https://inst.eecs.berkeley.edu/~ee127a/book/login/1_svd_apps_image.html](https://inst.eecs.berkeley.edu/~ee127a/book/login/1_svd_apps_image.html)