Assignment #2
Due Feb. 26, 2020

After that date, homeworks should be put into the mbox of one of our TAs

(1) You are given the matrix

\[ A = \begin{pmatrix} 2 & -2 \\ 1 & 1 \end{pmatrix}. \]  

(a) Find the spectrum and the spectral radius of this matrix (show your work). Use then Matlab’s eig and abs routines to check your answers.

Note, with \( \sigma(A) \) the spectrum of the square matrix \( A \), the spectral radius \( \rho(A) \) is defined by

\[ \rho(A) := \max\{|\lambda| : \lambda \in \sigma(A)\}. \]

(b) Compute the 1-, 2-, and \( \infty \)- norms of \( A \) above (show your work). Use then Matlab’s norm routine to check your answers.

(c) Find the left singular vectors, right singular vectors, and the singular values of \( A \). Check that \( A \) as well as \( A' \) map right/left singular vectors to left/right singular vectors. Based on your findings above, write the SVD of \( A \), and compare it with Matlab’s command svd.

(2) Let \( A \) be a symmetric matrix.

(a) For a \( 6 \times 6 \) \( A \) of your choice, use Matlab in order to find the spectrum and eigenvectors of \( A \) and the spectrum and eigenvectors of \( A'A = A^2 \) (your matrix \( A \) must be non-singular and cannot have any zero entry). Based on that example, conjecture a general connection between the spectrum and the eigenvectors of symmetric \( A \) and the spectrum and eigenvectors of \( A'A \). You will need to assume that if \( \lambda \) is an e-value of \( A \), then \( -\lambda \) is not, so you will not cover in your conjecture every symmetric matrix. (The conjecture should be of the form ‘(\( \lambda, v \)) is an eigenpair of \( A'A \) if and only if .... is an eigenpair of \( A' \)). If you do not find any reasonable conjecture to make, run more examples. However, turn in the Matlab output of one of your tests only.)

(b) Prove your conjecture from (a). Note that there are two parts in the proof (the ‘if’ part and the ‘only if’ part).

(c) In view of the above, state a theorem that derives the 2-norm of a symmetric \( A \) from its spectrum. Check your theorem against the matrix

\[ \begin{pmatrix} -8 & 144 \\ 144 & -92 \end{pmatrix}. \]

(d) Show that your theorem in (c) does not hold, in general, for matrices that are not symmetric (for that, you simply need to provide an example).

(e) What can you, thus, say about the singular values of a symmetric matrix? (Remember: singular values are the squareroots of the e-values of \( A'A \).)

(3) Let \( A \) be an invertible matrix.

(a) Prove that \( (\lambda, v) \) is eigenpair of \( A \) if and only if \( (1/\lambda, v) \) is an eigenpair of \( A^{-1} \) (check examples first if you feel confused).
(b) Use the claim in (a) in order to find a formula for $\|A^{-1}\|_2$ in terms of singular values of $A$. Explain.

(4) $QR$ factor the matrix $Z$ on page 76 of the book. You may use Matlab to this end, but must show all the intermediate results of your work. (So, you cannot use the $qr$ routine of Matlab.)

(5)
(a) Describe an efficient algorithm for computing the product $HA$, with $H$ a Householder matrix, and $A$ a general matrix. You may assume that $H$ is not given explicitly, and that, instead, the input on $H$ is the corresponding vector $w$. (Note: Multiplication of two square matrices of size $m$ requires $O(m^3)$ operations, despite of the fact that the number of entries is only $2m^2$. Your algorithm should compute $HA$ with $O(m^2)$ operations, which is best possible, since there are $m^2$ entries in $A$).

(b) Using (a) above, write a short code that efficiently solves a square linear system of equations using $QR$ factorization. Guideline: never compute any Householder matrix; simply save its vector $w$. After all, the only time the Householder matrix does anything, it is being multiplied by a vector or a matrix.

(c) Run your algorithm on two examples of your choice. The two matrices you choose should have at least order 5 each, and should be invertible.

(d) Estimate the complexity of your algorithm, i.e., the number of operations it uses as a function of the order of the system. Note: if your algorithm is designed correctly, the complexity is determined by the $QR$ factorization itself, and not by the subsequent need to solve the system.