Homework set 3 — APPM4720/5720, Spring 2016

Problem 1: Suppose that $\mathbf{A}$ is a real $n \times n$ symmetric matrix with eigenpairs $\{\lambda_j, \mathbf{v}_j\}_{j=1}^{n}$, ordered so that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$. Define a sequence of vectors via $\mathbf{x}_0 = \mathtt{randn}(n,1)$, and then $\mathbf{x}_p = \mathbf{A}\mathbf{x}_{p-1}$, so that $\mathbf{x}_p = \mathbf{A}^p\mathbf{x}_0$.

(a) Set $\beta = |\lambda_2|/|\lambda_1|$ and $\mathbf{y}_p = (1/\|\mathbf{x}_p\|)\,\mathbf{x}_p$. Assume that $\lambda_1 = 1$ and that $\beta < 1$. Prove that as $p \to \infty$, the vectors $\{\mathbf{y}_p\}$ converge either to $\mathbf{v}_1$ or to $-\mathbf{v}_1$.
(b) What is the speed of convergence of $\{\mathbf{y}_p\}$?
(c) Assume again that $\beta = |\lambda_2|/|\lambda_1| < 1$, but now drop the assumption that $\lambda_1 = 1$. Prove that your answers in (a) and (b) are still correct, with the exception that if λ_1 is negative, then it is the vector $(-1)^p\mathbf{y}_p$ that converges instead.

In solving this problem, you are allowed to use that when $\mathbf{x}_0$ is drawn from a Gaussian distribution (which is what the Matlab command `randn` does) it has a series expansion $\mathbf{x}_0 = \sum_{j=1}^{n} c_j\,\mathbf{v}_j$ where $c_j \neq 0$ with probability 1. (In fact, one can prove that each c_j is a random variable drawn from a normalized Gaussian distribution.)

Problem 2: The file `hw03p02.m` contains implementations of the Lanczos and Arnoldi procedures, and compares the accuracy obtained from these to the randomized SVD. But as you run the code, you will notice that while Lanczos and Arnoldi initially perform well, they quickly lose all accuracy. Lanczos fails fast, and Arnoldi fairly fast. The reason for this failure is the loss of orthonormality among the orthonormal vectors constructed. Modify the code to fix this problem. Once the subroutines that implement Lanczos and Arnoldi have been fixed, run the following numerical experiments:

(a) Pick some symmetric real matrix $\mathbf{A}$. Run a test that compares RSVD to the full SVD and to Lanczos.

(b) Pick some normal real matrix $\mathbf{A}$. Run a test that compares RSVD to the full SVD and to Arnoldi.

(c) Pick some non-normal real matrix $\mathbf{A}$. Run a test that compares RSVD to the full SVD and to Arnoldi.

For text matrices, you are welcome to use the matrices provided in the code, or some other matrices whose singular values decay that you find of interest. For each subproblem, hand in a print-out of the error graphs. Also include a description of how you edited the code, and a printout of the corrected Lanczos and Arnoldi codes.

Problem 3: Let $\mathbf{A}$ be an $m \times n$ matrix whose numerical rank is approximately k.

(a) Implement a code that computes the column ID with calling sequence

$$[J_{\rm s},\,\mathbf{Z}] = \mathtt{ID\_col}(\mathbf{A},k),$$

that produces an approximate rank-k factorization $\mathbf{A} \approx \mathbf{A}(:,J_{\rm s})\,\mathbf{Z}$. Use the algorithm based on the column pivoted QR factorization described in Figure 4.1 of the course notes. Hand in a print-out of your code.

(b) Implement a code that computes the CUR decomposition with calling sequence

$$[I_{\rm s},\,J_{\rm s},\,\mathbf{U}] = \mathtt{ID\_col}(\mathbf{A},k),$$

that produces a rank-k CUR $\mathbf{A} \approx \mathbf{A}(:,J_{\rm s})\,\mathbf{U}\,\mathbf{A}(I_{\rm s},:)$. Use the algorithm in Figure 4.4 of the course notes. Hand in a print-out of your code.

(c) *[Optional problem.]* Read Remark 4.6 in the course notes carefully. Then try to construct a matrix for which "Method 2" would either fail outright, or lead to a very numerically bad CUR. Hand in a description of the matrix, or code that generates it.