

Homework 1

If a correct solution is handed in before the deadline, two bonus points will be awarded to the final written exam. If a solution that is handed in before that date is not correct, it has to be redone, but the second time without yielding bonus points for the exam.

Solutions must be clearly written, and easy to follow. If not, they will not generate bonus points, and must be redone.

A written report with the solutions (hard-copy) should be handed in during the lecture or in the "homework mailbox" next to the mathematics studentexpedition on Lindstedsvägen 25. In addition to the report, you need to send me your source code by e-mail (eliasj at kth.se).

Deadline Homework 1: Nov. 14, 2014

1. Consider the following matrix:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & 2 & 9 \end{pmatrix}.$$

Plot the eigenvalue error as a function of iterate for the following algorithms in a semilog plot, and relate with the convergence theory.

- Power method with the starting vector $x_0^T = (1, 1, 1)$.
- Rayleigh quotient iteration with the starting vector $x_0^T = (1, 1, 1)$.
- Change the entry of the matrix in element row 1, column 3, to $a_{1,3} = 4$ and again run the Rayleigh quotient iteration with $x_0^T = (1, 1, 1)$. Explain why we obtain slower convergence in comparison to (b).

2. In this exercise we will investigate the performance of different versions of the Gram-Schmidt orthogonalization when it is combined with the Arnoldi method. Consider the matrix A constructed with the following command

$$A = \text{gallery}('wathen', 10, 10); \quad (1)$$

You may in this exercise use `eig(A)` as a reference/exact solution.



- (a) Modify the orthogonalization in `arnoldi.m` available from the course webpage. Apply it to (1) and generate the values in the following table, where `time` is the CPU-time and `orth` = $\|Q_m^T Q_m - I\|$ is an indicator of the orthogonality of the basis, and m the number of iterations in Arnoldi's method.

Use MATLAB-commands `tic` and `toc` to measure CPU-time.

	single GS		modified GS		double GS		double GS v2		triple GS	
	time	orth	time	orth	time	orth	time	orth	time	orth
$m = 5$										
$m = 10$										
$m = 20$										
$m = 50$										
$m = 100$										

- (b) Is one version "best" in this setting? If so, in what sense is it "best"? Also discuss the fairness of the comparison.

3. We shall here investigate a primitive variant of the Arnoldi method.

- (a) Let K_m be a matrix with the iterates of the power method, that is, let $K_m := [b, Ab/\|Ab\|, \dots, A^{m-1}b/\|A^{m-1}b\|] \in \mathbb{R}^{n \times m}$.

Show that the orthogonal projection (Galerkin projection) onto the Krylov subspace $\mathcal{K}_m(A, b) = \text{span}(b, Ab, \dots, A^{m-1}b)$ are solutions to the generalized eigenvalue problem

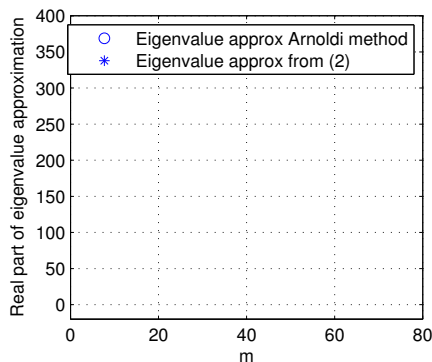
$$\mu K_m^T K_m w = K_m^T A K_m w \tag{2}$$

You may assume that $\mathcal{K}_m(A, b)$ has dimension m .

- (b) Solve (2) with `eig`. This directly gives a method which does not require any orthogonalization. Apply this method to (1) and compare with the Arnoldi method. Use a good orthogonalization procedure (not *double GS v2*). Generate the following figure for $m = 1, 2, \dots$. Use a random starting vector, with the same starting vector for all simulations.

The definition of K_n was modified on 2014-11-12. You are free to use the old or the new formulation

Hint: Recall that if V_m is an orthogonal basis of $\mathcal{K}_m(A, b)$, then the orthogonal projection are the eigenvalues of $V_m^T A V_m$. Note that there exists a non-singular matrix R such that $K_m = V_m R$. Show that $R^T R = K_m^T K_m$ by considering $V_m^T V_m$.



- (c) For what values of m do the results of the algorithms differ? Which approach is better? Why?



4. Download and load the matrix

<http://www.math.kth.se/na/SF2524/matber14/Bwedge.mat>

- (a) Accurate estimations of the eigenvalues are given in `B_eigvals`. Plot these and use the convergence theory for Arnoldi method for eigenvalue problems to
- identify which eigenvalues will be approximated first, and
 - estimate a convergence factor of the eigenvector angle (by estimating circles in a plot with the eigenvalues) corresponding to the first three eigenvalues that will be approximated.
- (b) Generate figures with the Ritz values for the Arnoldi method with a random starting vector, and verify and relate with (a) for $m = 2, 4, 8, 10, 20, 30, 40$.
- (c) Use the Arnoldi method to compute eigenvalues close to $\sigma = -11 + 2i$, with shift-and-invert. Carry out an LU-factorization of $C = B - \sigma I = LU$ and modify the reference implementation `arnoldi.m` such that in every step you need to solve one upper-triangular and one lower-triangular system of equations. Plot the approximations of B for $m = 2, 4, 8, 10, 20, 30, 40$ and relate to the eigenvalues of C^{-1} .

Use the command `load('Bwedge.mat')` to access the matrix B and eigenvalues.

Use the matlab command `[L,U,P]=lu(C)` to compute a (variant of a) LU-factorization. Note that this function computes a factorization $PC = LU$ such that $C \setminus z = L \setminus (U \setminus (P * z))$.

Only for PhD students taking the course *Numerical linear algebra*:

5. Exercise about restarting. Use the matrix `Bwedge.mat` in the simulations.

- (a) Implement an explicit restarting strategy for the Arnoldi method as follows. Take a linear combination (with unit weights) of the Ritz vectors generated by m iterations of the Arnoldi method as a new starting vector. Use the Ritz vectors corresponding to the k largest Ritz values. Extract the Ritz vectors as follows:

```
[Q,H]=arnoldi(A,b,m);
[V,D]=eig(H(1:end-1,1:end)); d=diag(D);
[Y,I]=sort(-abs(d));
ritz_vals=d(I(1:k));
ritz_vecs=Q(:,1:end-1)*V(:,I(1:k));
```

Carry out the experiments for $m = 10, k = 5$ and $m = 20, k = 10$ with 100 restarts. Plot the eigenvalue approximation as a function of restart. Does the restart strategy work in practice?

- (b) Use the reference code `arnupd.m` and `arnoldi_sorensen.m` which are implementations of an implicit restarting strategy. Run the algorithm with same restarting parameters as in (a) and explain the difference. Implicit restarting is in general to prefer over explicit restarting. Why?

The programs `arnoldi_sorensen.m` and `arnupd.m` are reference implementations of *Implicit application of polynomial filters in a k-step arnoldi method*, D. C. Sorensen, *SIAM J. Matrix Anal. Appl.* Vol. 13, No. 1, pp. 357-385, 1992. The programs are available from <http://www.math.kth.se/~eliasj/NLA/arnupd.m>