Homework 1

Information about homework:

- If correct solutions are handed in before the homework 1 deadline, bonus point will be awarded for the final exam. The deadline is visible in CANVAS Homework. If solutions to questions 2-6 handed in before that date are not correct, they have 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. It is highly recommended that the answers are done on a computer (for instance LaTeX or word). Task 2-4,6-8 can be done in groups of two (single person groups are allowed). Please submit, not later than midnight of the date of the homework 1 deadline:

  - written solutions (in the form of a PDF-file)
  - your matlab (or julia) programs for the problems

  The files should be uploaded in CANVAS → SF2524 → Assignments → Homework 1

- If the extended version of task 1 is solved before the *wiki-deadline*, 1 wiki-bonus will be awarded. If you collect z wiki-bonus points during the course, the grade limits for grade A and B will be reduced by z points. You submit your wiki tasks by specifying questions numbers (before wiki deadline) in
  CANVAS → SF2524 → Assignments → Wiki part homework 1

Active learning workspace: http://dhcp-41-104.math.kth.se/~jarl/active_learning/active_learning.php

1. Create problems and solutions on the course training wiki before the wiki deadline. This task is optional for SF2524, but can give you more bonus points. See instructions here:

   CANVAS → SF2524 → Modules → Homework & Bonus points rules

   Note in particular: Please do not solve more problems than you post.

   For this homework you can post or answer wiki problems in Block 1 or Block 2.

2. Consider the following matrix:

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

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

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

(a) Power method with the starting vector $x_0^T = (1,1,1)$.

(b) Rayleigh qoutient iteration with the starting vector $x_0^T = (1,1,1)$.

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

3. 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

```
rand('seed',0); A=gallery('wathen',nn,nn);      (1)
```

(a) Modify the orthogonalization in `arnoldi.m` available from the course we page. 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.
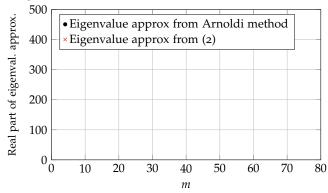
Use tic-toc to measure performance in MATLAB. In order to get reliable estimates, you may need the tricks described here: http://se.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html.

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

Select nn$\geq$ 10 such that the maximum computation time for $m = 100$ is approximately 10 minutes on your computer.

(b) Interpret the result in (a). Is one version "best" in this setting? If so, in what sense is it "best"? Discuss the fairness of the comparison.

4. We shall here investigate a primitive variant of the Arnoldi method. Let K_m be a matrix with the iterates of the power method, that is, let $K_m := \left[b, Ab/\|Ab\|, \ldots, A^{m-1}b/\|A^{m-1}b\| \right] \in \mathbb{R}^{n \times m}$.

Hint: See wiki 1-40.

(a) We will let eigenvalues of A be approximated by eigenvalues of the matrix $(K_m^T K_m)^{-1} K_m^T A K_m$:

$$\mu w = (K_m^T K_m)^{-1} K_m^T A K_m w \qquad (2)$$

You may assume that $K_m^T K_m$ is invertible. Prove that the approximation (2) is identical to the approximation generated by Arnoldi's method for eigenvalue problem. You may use any theorem in the course.

(b) By solving (2) directly with `eig`, we have a method which requires only computation of $K_m^T K_m$ and $K_m^T A K_M$, and does not require any orthogonalization. Note that the matrix in the right-hand side is $m \times m$ which is a small matrix and you can use `eig`. Apply this method to (1) and compare with the Arnoldi method. Use double GS as orthogonalization. Generate the following figure for $m = 1, 2, \ldots$: Use a random starting vector, with the same starting vector for all simulations.

Note that $K_m^T A K_m$ has m eigenvalues, so the plot will consist of several curves/points.

(c) Interpret the result of (a)-(b). What do we expect in exact arithmetic? What do we observe? Which approach is better? Why?

5. Do the quiz "Lanczos algorithm" on CANVAS. You can do the quiz as many times as you like. This task is individual and should not be done as a group. No reply in the PDF-file is required for this problem.

6. Download and load the matrix

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

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

(a) The eigenvalues are given in the variable `B_eigvals`. Plot the eigenvalues and mark the eigenvalues you will have the fastest convergence with Arnoldi's method.

(b) In the same figures as in (a), indicate circles and estimate a convergence factor. The convergence factor is here meant $\alpha < 1$ such that the error $\sim \alpha^m$.

Julia users: see file instructions CANVAS → SF2524 → Modules → Julia instructions

(c) Generate figures with the Ritz values for the Arnoldi method, with double GS for $m = 2, 4, 8, 10, 20, 30, 40$. Clearly indicate which eigenvalue you expect fast convergence based on (a). How many iterations are needed in order to get an eigenvalue error approximately 10^{-10}

(d) The matrix has an eigenvalue close to $-9.8 + 2i$, which can be found in the `B_eigvals` variable. Implement shift and invert Arnoldi method. Run the algorithm several times for different shifts σ in the table below, and inspect the error after m iterations. Complete the table with the eigenvalue errors: the difference between the exact eigenvalue and the closest computed approximation ($\tilde{\lambda} = \sigma + 1/\mu$ where μ is the Ritz value). The prefilled values in the Standard AM column, refers to approximation errors obtained from the plain Arnoldi method without shift-and-invert. Which situation converges faster, and why?

MATLAB hint: If you want to find the element in a vector xv which is closest to a given value z you can use `min` with two return parameters:
```
>> [~,I]=min(abs(xv-z));
>> xclosest=xv(I);
```

	Standard AM	$\sigma = -10$	$\sigma = -7 + 2i$	$\sigma = -9.8 + 1.5i$
$m = 10$	2.5			
$m = 20$	0.6			
$m = 30$	$2 \cdot 10^{-4}$			

Only for *SF3580 Numerical linear algebra*.

7. 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?

8. SVD: Singular value decomposition. If you have not heard of SVD, you may need to look it up, in particular the Eckart-Young theorem.

   (a) What are the relationship between the singular values of the matrix $A \in \mathbb{R}^{n \times m}$ and the eigenvalues and eigenvectors of

   $$B = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

   (b) Download the movie `india_driving1` from https://kth.instructure.com/groups/97069/files. Reshape the movie to a matrix. Compute the SVD (with `svd(A)`) and construct the best rank one approximation
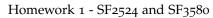
   $$\tilde{A} = u_1 s_1 v_1^T$$

   According to Eckart-Young theorem, this corresponds to the largest singular value and corresponding vector. Visualize and interpret the first column of $\tilde{A}$ as an image. Explain (briefly) why this is image looks this way.

   (c) Download the larger movie file `market`. Reshape the movie to a matrix. The `svd` does not work on this matrix (on most computers) since the matrix is too large and the `svd`-algorithm runs

Note that even if the eigenvector of B is normalized, the two components correspond to singular vectors need to be normalized in a post-processing step.

See CANVAS "Julia instructions" for how to load and manipulate videos frames in Julia.

---

out of memory. Use (a) and the Lanczos method to compute the singular vector and solve visualize as in (b).