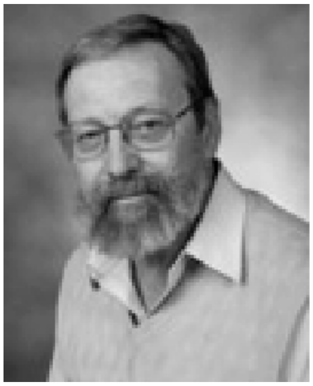
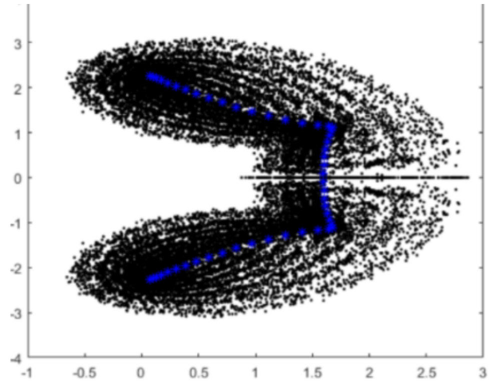
Numerical Linear Algebra **Assignment 2**


This text describes Assignment 2 for the course Numerical Linear Algebra (MSc). You are encouraged to work on this assignment in couples and to hand in together as a team, although this is not mandatory. Students who work together as a team get the same grade. For Assignment 2 you can choose again to work alone or with someone else, possibly a different person than for Assignment 1. Each assignment counts for 10% towards your final grade.

Warning: to hand in together with somebody who did not contribute to the assignment with the intention to supply this person with a sufficient grade is considered fraudulent behavior that goes against academic integrity.

So, please do not be tempted to "work together" with someone who either is not able or not willing to contribute to the solution of the assigment in a similar amount as yourself!

Uploading a self-contained video presentation

You are asked to compose and edit a coherent video in which you address the issues and questions that are raised in the assignment text. Perhaps the best way to do so is to consider it as a presentation of your team in front of your fellow students in the course. You are invited to use LaTeX slides in combination with demonstrations of your computer codes and discussions of the coding itself, in order to show you understand the theoretical exercises, that you did the coding, and that the codes are actually functioning and producing numbers and pictures. Try to make the video a smooth narrative, by drawing up a scenario beforehand and by rehearsing. The expected length of the video is about 20 minutes. You are allowed to do editing in any way you like, gluing together smaller parts if desired. If you work as a couple, make sure that both of you have equal roles in the video, like being visible and talking about the work approximately equally much. Avoid that one person does the easy exercises only.

Disclaimer: As much as I have tried to make this text accurate and correct, it may well be that typo's, smaller errors, and larger errors have entered it, because the present times are rather demanding. If you have doubts about anything in the text, please let me know as quickly as possible in order to prevent potential damage as much as possible. Many thanks!

And have fun! ☺ Jan Brandts

Krylov Subspaces for Eigenvalue Problems

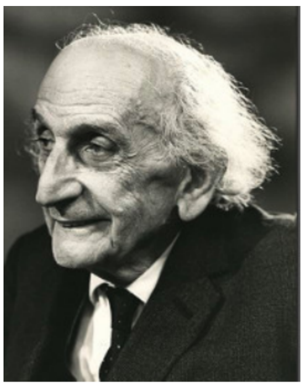
In this assignment you will implement the Lanczos and the Arnoldi method for eigenvalues.

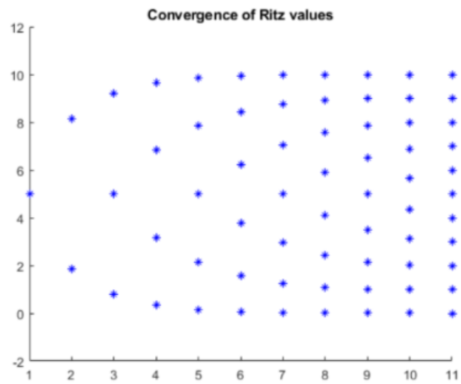
1. The Lanczos method and the Arnoldi method

Let $A \in \mathbb{R}^{n \times n}$ and $0 \neq v \in \mathbb{R}^n$. The Arnoldi method computes an orthonormal basis for the Krylov subspace

$$\mathcal{K}^k(A, v) = \operatorname{span}\{v, Av, A^2v, \ldots, A^{k-1}v\}. \tag{1}$$

This can for instance be done by Classical Gram-Schmidt (CGS), or equally expensive by Modified Gram-Schmidt (MGS) or for about twice the costs by Householder reflections (HHR).





Cornelius Lanczos (1873-1974) and Walter Ritz (1879-1909)

MGS and HHR may still exhibit substantial differences in finite precision arithmetic. When $A^\top = A$, orthogonality is automatic to all but the two previously computed basis vectors, and orthogonalization against (only) these last two behaves similarly for MGS and HHR.

The Lanczos method yields in step $k < n$ a matrix $V_{k+1} \in \mathbb{R}^{n \times (k+1)}$ with orthonormal columns and a tridiagonal matrix $T_{k+1,k}$ such that $AV_k = V_{k+1}T_{k+1,k}$ and

$$V_{k+1}^\top V_{k+1} = I, \quad \text{and} \quad V_k^\top A V_k = T_{k,k}.$$

We will compute the Ritz values and Ritz vectors. Then we study their residuals $r_1, \ldots, r_k$ in relation to the uniform upper bound $|t_{k+1,k}|$ that was proved for their norms.

Exercise 1: Let $A^\top = A \in \mathbb{R}^{n \times n}$. Write a main program that applies the Lanczos method to A with start vector $v \neq 0$. At each step k, it should display the following data[a] in the form of a $2 \times (k+1)$ matrix

$$\begin{array}{|c|cccc|} \hline k & \mu_1 & \mu_2 & \cdots & \mu_k \\ \hline |t_{k+1,k}| & \|r_1\| & \|r_2\| & \cdots & \|r_k\| \\ \hline \end{array} \tag{2}$$

where $\mu_1 \leq \ldots \leq \mu_k$ are the Ritz values and r_j is the residual for μ_j and its corresponding Ritz vector. After n steps, a plot should be made with the iteration number k at the horizontal axis, and a marker (i.e. asterisk) for each Ritz value on the vertical axis. You may use built-in commands to solve the eigenvalue problem for $T_{k,k}$ in each step.

[a] note that $t_{n+1,n}$ does not exist; display 0 instead.

Exercise 2: Apply your code to the diagonal matrix with eigenvalues $0, 1, \ldots, 10$ and with start vector $v = (1, \cdots, 1)$. Step by step during the iteration, comment on the matrices in (2) that are displayed, and on the picture displayed at the end. Repeat the experiment with $v = (2^0, 2^1, \cdots, 2^{10})$. Comment on the differences with the previous start vector.

Exercise 3: Because for symmetric A the eigenvectors of $T_{k,k}$ are orthonormal, a much stronger relation than $\|r_j\| \leq |t_{k+1,k}|$ for $j = 1, \ldots, k$ can be proved for the sizes of the residuals at a given step, a result that moreover does not involve these orthonormal eigenvectors explicitly. See if you find it by studying the numerical results, then confirm it by doing another experiment with another matrix, and then try to prove it.

2. Perturbation theory, backward stability, and pseuso-spectra

We will now turn our attention to the Arnoldi method. To illustrate how badly the Ritz values may approximate eigenvalues we use a test matrix from the Matlab gallery. See `help gallery` for a complete list of test matrices and a brief description. Set `>> G = gallery('grcar',50)` to create the so-called Grcar matrix G of dimension 50×50. You may look at some Grcar matrices of smaller dimension to understand their definition.

The Grcar matrices are named after Joseph Grcar. In particular, Grcar is not an abbreviation of some kind, as was originally thought by your lecturer[a]. On November 6, 2018, Grcar ran for candidate for the California Assembly in the General Election, with slogans Put Alameda County First! and Make California Great Again! In the 2019 NLA course a student remarked that he wasn't elected. He will surely be remembered for his matrix ☺

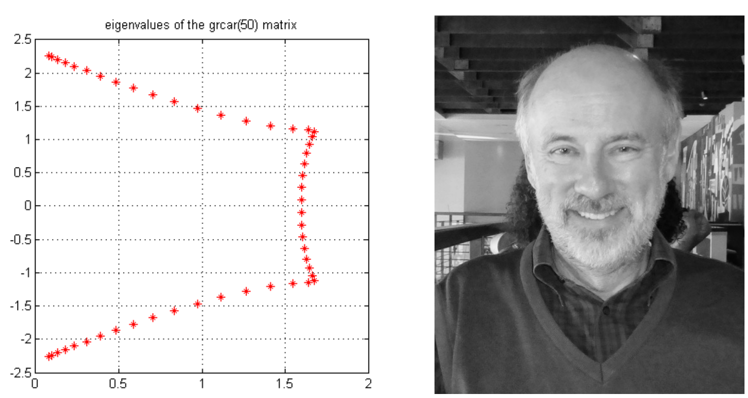


Figure 1. Joseph Grcar (b. 1951) and the spectrum of his matrix of size 50×50.

[a] A misunderstanding that was resolved many years later when your lecturer went to a conference and met a man called Joe who, quite unusually, walked around with the name of a matrix on his badge ☺

Not only Grcar, but also G is real. Hence its eigenvalues and Ritz values come in complex conjugate pairs. Instead of using $\mathbb{C} \times \{1, \ldots, k\}$ to visualize the convergence of Ritz values (which is nontrivial) we will use a Matlab movie.

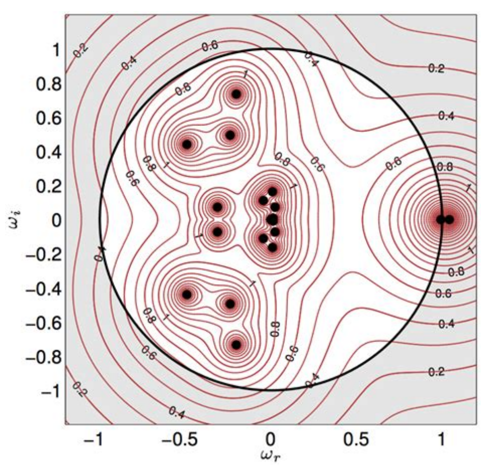
Remark: Relevant Matlab commands for Exercise 4 are movie, getframe, axis. In case you are not well acquainted with Matlab's movie-options, you may also produce a poor man's version by hammering on the enter-key at a certain frequency and show a number of consecutive plots. You won't be punished in your grade (hammering is sufficient punishment☺)

Exercise 4: Make a movie that in frame k shows the k Ritz values in blue derived from the Arnoldi method on top of the picture in Figure 1 containing the exact eigenvalues for reference. Thus, going from one frame to the next, the Ritz values from the previous step should be erased (or overwritten by white asterisks) and the ones from the current step should appear, all with Figure 1 as background. Show two frames per second, so that the whole movie plays within one minute. Keep the domain during the whole movie fixed at $-1 \leq \Re(z) \leq 3$ and $-4 \leq \mathrm{Im}(z) \leq 4$ (in $\mathbb{C}$). Use the all-ones vector as start vector. What is your opinion on the convergence properties of the Ritz values in this notorious example?

Given a matrix A and an $\varepsilon \geq 0$, the set

$$\sigma_\varepsilon(A) = \{\sigma(A+E) \mid \|E\| \leq \varepsilon\} \tag{3}$$

is called the ε-pseudospectrum[1] of A. If A is normal, $\sigma_\varepsilon(A)$ is simply the union of the discs with radius ε around the eigenvalues of A. Obviously, $\sigma_a(A) \subset \sigma_b(A)$ whenever $a \leq b$. The below picture gives an impression of these nested sets for various sizes of ε (matrix unknown).



Recall that if (μ, v) is a Ritz pair produced by the Arnoldi method with residual r, this Ritz pair is an exact eigenpair of the matrix $G+E$ where $E = -rv^*$, and $\|E\| = \|r\|$. Therefore, we have that

$$\mu \in \sigma_{\|r\|}(G) \tag{4}$$

hence the residual size of a Ritz pair shows in which ε-pseusospectrum μ lies.

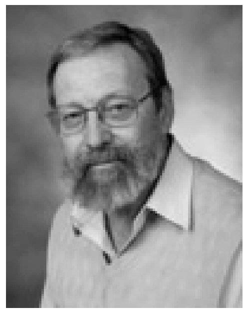
Exercise 5: For each $\nu \in \{1, 2, \ldots, 10\}$, use Matlab to compute the eigenvalues of 100 matrices $G + 10^{-\nu}uv^\top$, where G is the 50×50 Grcar matrix above, and u and v with $\|v\| = 1 = \|u\|$ are randomly generated. Superimpose all 5.000 computed eigenvalues as black dots in one plot. After that (not before or they might be covered by black dots) add the exact eigenvalues as red asterisks. This gives you ten plots, each one an approximation of $\sigma_\varepsilon(G)$ with $\varepsilon = 10^{-\nu}$. What do these plots tell you about the potential success of eigenvalue algorithms that try to approximate eigenvalues of G, and of Arnoldi in particular?

We will now try to improve the Arnoldi method, by manipulating its start vector.

[1]See the website pseudospectra gateway of Nick Trefethen en Mark Embree et. al.

3. The Implicitly Restarted Arnoldi method

The Implicitly Restarted Arnoldi method is generally known (Ireland excluded) by its abbreviation as IRA. It was developed by Dan Sorensen and Rich Lehoucq in 1996.




Danny Sorensen (b. 1947) and Rich Lehoucq (b. 1966)

The central idea of their work is to extract, without using the matrix A again, from a given Arnoldi factorization

$$AV_k = V_{k+1}H_{k+1,k} \tag{5}$$

for the Krylov space $\mathcal{K}^k(A,v)$ the Arnoldi factorization

$$AW_{k-\ell} = W_{k+1-\ell}\hat{H}_{k+1-\ell,k-\ell} \tag{6}$$

for the Krylov space $\mathcal{K}^{k-\ell}(A,p(A)v)$ for a given polynomial p of degree ℓ. In case $p(x)=x-\mu$ where $\mu\in\mathbb{C}$ is an eigenvalue of $H_{k,k}$, then $\sigma(\hat{H}_{k-1,k-1})=\sigma(H_{k,k})\setminus\{\mu\}$. In other words, the Ritz value μ has been removed from the original Arnoldi factorization.

For $\ell=1$, the details of the central idea are summarized in the following theorem.

Theorem. *For any given $\mu\in\mathbb{C}$, let $H_{k+1,k}-\mu I_{k+1,k}=Q_{k+1,k}R_{k,k}$ be a QR-decomposition, and set*

$$W_k = V_{k+1}Q_{k+1,k}. \tag{7}$$

Then W_k contains an orthonormal basis for $\mathcal{K}^k(A,\hat{v}_1)$, where

$$\hat{v}_1=\tilde{v}/\|\tilde{v}\| \quad\text{and}\quad \tilde{v}=(A-\mu I)v. \tag{8}$$

Moreover,

$$AW_{k-1}=W_k\hat{H}_{k,k-1} \quad\text{where}\quad \hat{H}_{k,k-1}=R_{k,k}Q_{k,k-1}+\mu I_{k,k-1}, \tag{9}$$

is the Arnoldi factorization corresponding to $\mathcal{K}^{k-1}(A,\hat{v}_1)$.

Proof. Because $Q_{k+1,k}$ is upper Hessenberg, so is $\hat{H}_{k,k-1}$. Furthermore, since $Q_{k+1,k}$ is upper Hessenberg, we also have the somewhat surprising equality

$$I_{k+1,k}Q_{k,k-1}=Q_{k+1,k}I_{k,k-1}. \tag{10}$$

Using this, it can be verified using the various definitions above, that

$$\begin{aligned} AW_{k-1} &= AV_kQ_{k,k-1}=V_{k+1}H_{k+1,k}Q_{k,k-1}=V_{k+1}(\mu I_{k+1,k}+Q_{k+1,k}R_{k,k})Q_{k,k-1}\\ &= V_{k+1}Q_{k+1,k}(\mu I_{k,k-1}+R_{k,k}Q_{k,k-1})=W_k\hat{H}_{k,k-1}. \end{aligned} \tag{11}$$

Since W_k is orthogonal and $\hat{H}_{k,k-1}$ is upper Hessenberg, we have that (9) is an Arnoldi decomposition. The equality

$$W_k e_1 = V_k Q_{k+1,k} \frac{R_{k,k} e_1}{r_{11}} = \frac{1}{r_{11}} V_k (H_k - \mu I) e_1 = \frac{1}{r_{11}} (AV_k - \mu V_k) e_1 = \hat{v}_1 \tag{12}$$

shows that W_k contains the orthonormal basis for $\mathcal{K}^k(A, \hat{v}_1)$. □

Exercise 6: Prove that if $\mu \in \sigma(H_{k,k})$, then $\sigma(\hat{H}_{k-1,k-1}) = \sigma(H_{k,k}) \setminus \{\mu\}$. Hint: see also Exercise 3(c-d) of Week 8.

The result of Exercise 6 confirms the relevance of the main idea behind IRA. If μ is an unwanted Ritz value, it can be removed without losing the other Ritz values.

Exercise 7: Write a function "[V,H] = FilterAway(mu,V,H)" which overwrites the given Arnoldi factorization (5) with start vector v with the new (but $\ell = 1$ smaller in size) Arnoldi factorization (6) with start vector $(A - \mu I)v$ using Theorem 1. Next, write a function "[V,H] = ExtendArnoldi(A,V,H)" which adds one additional iteration to a given Arnoldi factorization. Finally, write a function "L = ListRitzData(H)" whose output L, as in Exercise 1, contains the current Ritz values together with their residual norms. Test these functions properly. In particular, confirm the result of Exercise 6.

Exercise 8: Using your codes from Exercise 7, implement the IRA in Matlab in the following interactive way. At each moment, the user should be given the following choice:

- EXTEND: perform one more iteration;
- FILTER: filter away a subset (indices provided by the user) of the current Ritz values;
- EXIT: end all computations and return the current Arnoldi factorization.

Relevant Matlab commands in this exercise are: input, disp, pause.

As a simple test case for the implicit restarts, we briefly return to the context of Exercise 2.

Exercise 9: Run the code of Exercise 8 on the matrix A from Exercise 2 with start vector $v = (2^0, 2^1, \ldots, 2^{10})$. First, only use EXTEND, to basically repeat the second experiment of Exercise 3. For each $k \in \{1, \ldots, 10\}$ write down the smallest eigenvalue of the $k \times k$ matrix $T_{k,k} = H_{k,k}$, which gives a sequence of approximations of the smallest eigenvalue 0 of A. You may wish to verify if the results are, in fact, the same as in Exercise 2 (they should!☺)

Now, run the code of Exercise 8 from scratch and focus again on the smallest eigenvalue. EXTEND until you have computed the eigenvalues of $T_{4,4}$. Repeat the following two actions:

- FILTER away the unwanted[a] approximate eigenpair with the smallest residual;
- EXTEND the factorization back to size 4×4.

After each repetition, write down the smallest eigenvalue of the size 4×4 factorization. How many FILTER/EXPAND pairs are needed to approximate the smallest eigenvalue approximately equally well as after ten steps (i.e. $T_{10,10}$) without any filtering? Comment on this both in terms of accuracy and efficiency of the implicit restart strategy.

[a]By this we mean: not the smallest (=wanted) eigenvalue, which may well have the smallest residual!

4 Filtering away complex conjugate eigenpairs

Note that if you filter away only one of a complex conjugate pair of Ritz values, you end up with a complex Arnoldi factorization. If you then filter away its complex conjugate, you can choose to end up again with real matrices, just as in the square case that we outline now.

Observation. Consider the QR-algorithm for a square unreduced upper Hessenberg matrix H and apply two consecutive steps with shifts μ_1 and μ_2,

$$H - \mu_1 I = Q_1 R_1, \quad H_1 = R_1 Q_1 + \mu_1 I, \quad H_1 - \mu_2 I = Q_2 R_2, \quad H_2 = R_2 Q_2 + \mu_2 I. \tag{13}$$

Then some manipulations with the above definitions show that $H_2 = Q_2^* Q_1^* H Q_1 Q_2$, and

$$H^2 - (\mu_1 + \mu_2) H + \mu_1 \mu_2 I = Q_1 Q_2 R_2 R_1. \tag{14}$$

If $\mu_2 = \overline{\mu}_1$ then the matrix in the left-hand side of (14) is a real matrix. Thus, it has a real QR-decomposition QR. Hence, there exist a diagonal unitary matrix Z such that $(Q_1 Q_2) Z = Q$ and $\overline{Z}(R_2 R_1) = R$. Defining H_2 alternatively as $H_2 = Q^* H Q$ then results in a real matrix. Note that Q and $Q_2 Q_1 Z$ only differ in the way their columns are scaled to norm one. This is non-essential, as they are supposed to be eigenvector approximations.

A similar strategy to avoid complex arithmetic can be applied in the none-square case.

Exercise 10: Generalize Theorem 1 such that a complex conjugate Ritz pair is filtered away similarly as in the square case, and with a real resulting Arnoldi factorization, one whose length has now been reduced by two instead of one. Be very careful with manipulating the non-square matrices, for instance by making pictures of the rectangular matrices and doing numerical tests on small examples to make sure everything you do makes sense! ☺

Write a function "[V,H] = FilterAway2(mu,V,H)" that implements your above result, overwriting the given Arnoldi factorization (5) with start vector v with the new (but $\ell = 2$ smaller in size) Arnoldi factorization (6) with new start vector $(A - \mu I)(A - \overline{\mu} I)v$. Test it!