Homework 2

If correct solutions are handed in before the homework 1 deadline, and canvas peer-review is done correctly, bonus point will be awarded for the final exam. If an incomplete or incorrect solution is handed in, you will need to redo the homework, and might obtain partial bonus.
It is highly recommended that the answers are done on a computer (for instance LaTeX or word). All tasks should be done in groups of two (single person groups are allowed). Please submit, by the deadline specified in CANVAS

- written solutions (in the form of a PDF-file). CANVAS → SF2524 → Assignments → Homework 2
- your matlab (or julia) programs for the problems. CANVAS → SF2524 → Assignments → Homework 2 sourcecode

---

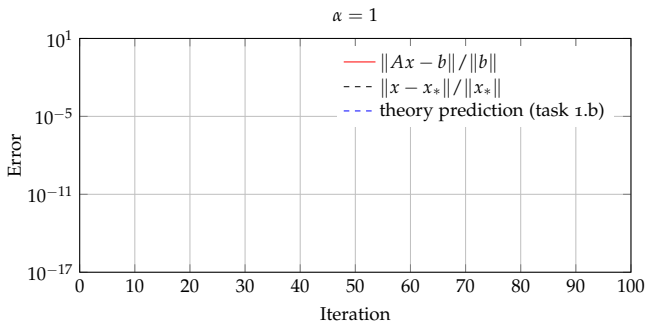
0. Specify here your AI usage level (0-3) and when applicable further explanation. The level can also be specified per solution.
See CANVAS → SF2524 → "Can I use generative AI?"

1. Implement GMRES (Generalized Minimum Residual method) based on the Arnoldi method, that you developed in Homework 1. Use double Gram-Schmidt for the orthogonalization. Consider the linear system $Ax = b$, where A and b are generated by

```
alpha=5;  n=100; rand('state',5);
A = sprand(n,n,0.5);
A = A + alpha*speye(n); A=A/norm(A,1);
b = rand(n,1);
```

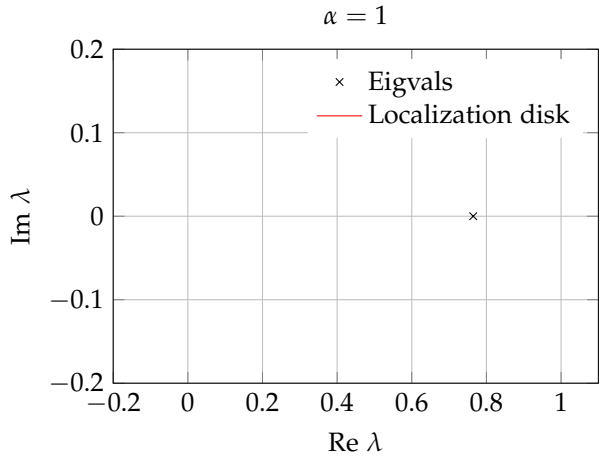
For all homeworks: Use the same x-axis and y-axis styles and limits as the provided figures unless you have good reasons to change it; otherwise we will normally mark it as incorrect. Also use the same markers and line styles. You may add other figures to explain your solutions/reasoning.

(a) Plot the norm of the relative error $\|x_* - x_m\|/\|x_*\|$ as a function of iteration as well as the relative residual norm $\|Ax_m - b\|/\|b\|$ as a function of iteration. Use `semilogy`. You may in this exercise use `A\b` as an exact solution. Generate figures for the values $\alpha = 1, 5, 10, 100$. For example for $\alpha = 1$:



The matlab backslash-command is based on extremely optimized LU-factorizations (or sometimes Cholesky factorizations or Cholesky decompositions).

(b) Plot the eigenvalues with `plot(eig(full(A)),'*')` for all choices of α in (a) and provide provide a bound for the convergence factor, using the single localization disk (you may choose if you want to use the outlier version or not). See example for $\alpha = 1$ below. Explain and relate the observed convergence to the convergence theory by plotting the estimated convergence factor (predicted by the eigenvalues) in convergence figures as in (a), in the same figures as (a) or separate figures.



(c) Generate the following table, where `resnorm`$=\|Ax - b\|_2$ and `time` is the CPU-time and m is the number of iterations. Make the corresponding simulations for the backslash operator \. Make tables for $\alpha = 1$ and $\alpha = 100$.

You may want to extend/modify the table with more rows/columns to support your claim, if your computer allows it.

GMRES

	$n = 200$		$n = 500$		$n = 1000$	
	resnorm	time	resnorm	time	resnorm	time
$m = 5$						
$m = 10$						
$m = 20$						
$m = 50$						
$m = 100$						

Backslash

	$n = 200$		$n = 500$		$n = 1000$	
	resnorm	time	resnorm	time	resnorm	time

(d) Suppose we are in a situation where it is sufficient to compute a solution to accuracy (relative residual norm) 10^{-5}. Is GMRES better than the backslash operator in this situation? Depending on your specific computer, you may want to extend/modify the table in c) to support your claim.

2. Consider the A-matrix and b-vector generated in this way.

```
n=100;
e = ones(n,1);
A = -spdiags([e -6*e e], -1:1, n, n)
b = kron(ones(n/2,1),[0;1])+ones(n,1);
```

We will compare three different methods to solve $Ax = b$.

- Method 1) CG
- Method 2) GMRES
- Method 3) CG with least squares: Form an approximation $x = Xz$, where z is

$$z = \underset{z \in \mathbb{R}^p}{\operatorname{argmin}} \|AXz - b\| \qquad (\dagger)$$

and $X = [x_1, \ldots, x_p]$ is a matrix with the iterates generated by CG. The minimization problem ($\dagger$) is a linear least squares problem and should be solved by using the normal equations (not backslash on a rectangular matrix). See 0.3 in background.pdf.

Method 3 has nothing to do with CGNE.

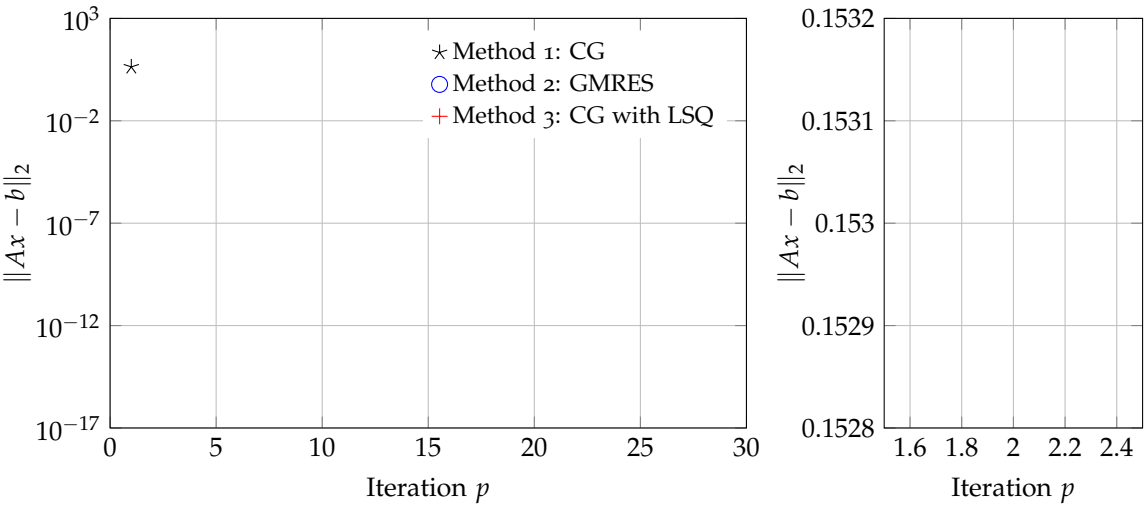
a) Generate illustrative figures with iteration p on the x-axis, and the residual norm $\|Ax - b\|$ on the y-axis. Use the standard 2-norm for all plots. The properties should be visible when using scales and markers in the example figures below.

Note that the first figure is in semilogy scale and the second figure is in regular scale on a very specific interval that only shows one iteration.

b) Interpretation 1: In theory, two of the methods should be identical. Which ones? Why? Relate to the definitions of method 1,2, and properties of method 3.

c) Interpretation 2: Relate (b) to what we see in practice? Which method / methods has /have lower y-value? Why?

d) Which method is the best for this problem in a realistic situation with a corresponding larger matrix?



3. Summarize in your own words (2-5 sentences for each video):
   (a) 201 CPU-socket application and iterative vs direct methods
   (b) 215 GMRES convergence single outlier
   (c) 216 Preconditioning for GMRES (part 1)
   (d) 234 Preconditioning CG part 1
   (e) 235 Preconditioning CG part 2
   (f) 250 BiCG method derivation
   (g) 2XX a video from this block that you can choose

Reminder: See slides from lecture 1 for usage of AI tools.

Note that several of the videos above are also the material for corresponding CANVAS quizzes.

4. Suppose A is a real symmetric matrix with eigenvalues $10, 10.5$ and 100 other eigenvalues in the interval

$$I = [2, 3].$$

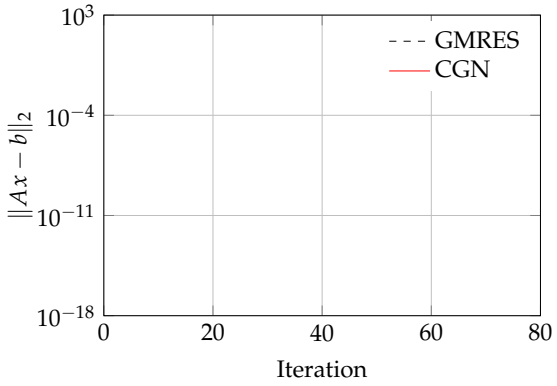
Prove a bound on how many steps of CG must be carried out in order to reduce the error (measured in $\|Ax_m - b\|_{A^{-1}} = \|x_m - x_*\|_A$) by a factor 10^7. You may assume exact arithmetic and that no premature breakdown occurs. Use any result in the course.

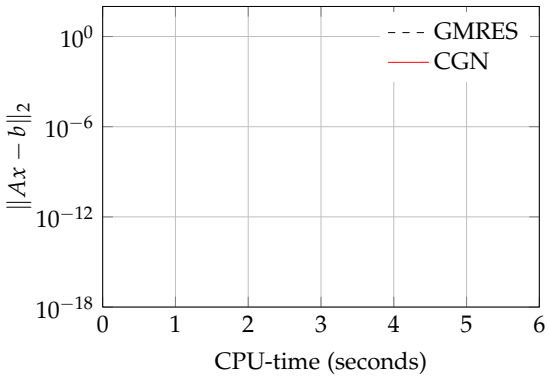
5. In the following problem use the mat-file `cgn_illustration.mat` containing both an A-matrix and a right-hand side vector b.

   (a) Compare GMRES and CGN for this problem by completing following two figures. Note that both figures illustrate the same simulation, but with different x-axis. The second figure highly depends on the computing environment, so it may be necessary to adjust the limits of the x-axis (increase or decrease the range of the time-axis). Try to make your implementation of CGN as efficient as possible. In particular, it should only contain *two matrix vector products involving the A-matrix per iteration*.

Hint: It is important and a bit difficult to do timing correctly. The second figure can be generated using bookkeeping of time-stamps: https://play.kth.se/media/t/0_o6mmqivi. In Julia you can use `Int(time_ns())`, see timepoint 04:00 in the video. See video 202a, 202b.
You can find data files here: (the course-round independent canvas page) https://canvas.kth.se/courses/59848/files

If you get zig-zagging in the CPU-time figure, watch the bookkeeping video above.





   (b) Interpret the result in (a) using the convergence theory for CGN and GMRES.

6. Download the files
   - `heatsink_exercise_refX.mat`
   - `heatsink_precond_refX.mat`
   - `heatsink_build_matrices.m`
   - `plot_heatsink.m`
   - `pcg_v1.m`

   Start with $X = 1$.

   Problem (b) is optional, but recommended for MATLAB users. (I may provide Julia plottig instructions later.)

Julia users: Use corresponding jl-files. Currently there is no plot_heatsink.jl, but it is not needed to solve the task. However, (optional) you can use `export_heatsink.jl` which exports the solution to a vtu-file that can be opened in tools such as paraview. (In paraview select "color by" temperature to actually see the solution.)

(a) What is the computation time for one matrix vector product of A generated by `heatsink_build_matrices` for model $X = 1$ and model $X = 2$? Carry out the simulation 10 times (or more) and form an average to get accurate timing.

(b) Use the plot functionionality `plot_heatsink` and visualize the heatsink. Give two heat profiles: Set the solution vector to $x = 293 * ones(n, 1)$ and the x-vector you get when you run `heatsink_build_matrices`. (Update 2025-11-20: The images will look very similar, which was not originally intended.)

(c) What is the computation time for computing backslash as in `heatsink_build_matrices`.

(d) Load the preconditioner files. Compute for 10 random choices of a vector z:

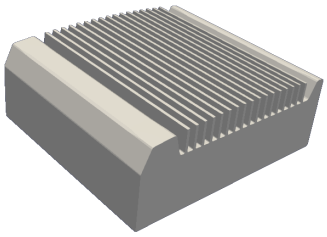
```
norm(T2\(T1\(A*z))-z)/norm(A*z)
```

Correspondingly for a diagonal precondioner $D = diag(A)$ with

```
D=spdiags(diag(A),0,n,n); norm(D\(A*z)-z)/norm(A*z);
```

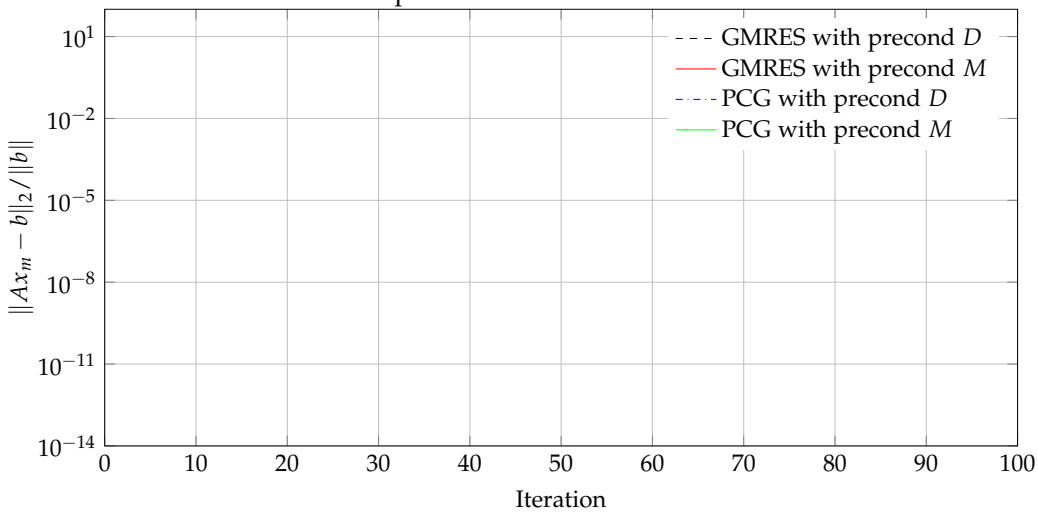
Based on this, is the diagonal preconditioner $A^{-1} \approx D^{-1}$ more accurate than $M^{-1} = T_2^{-1}T_1^{-1} \approx A^{-1}$? Carry out computation time comparison as in (a), for the preconditioner. Which one is faster?

(e) Adapt your GMRES implementation, and incorporate left preconditioning. Adapt pcg, based on the template code, with $M^{-1} = T_2^{-1}T_1^{-1}$. Use both the diagonal and the provided preconditioners. For full points, the pcg implementation needs to be improved to not carry out unnecessary matrix vector products. Carry out 100 iterations. Make a figure like below. Which one is best? For what accuracy is the iterative method a better idea than backslash in terms of computation time?



A FEM model of the Neosys Nuvo 7000-LP computer with embedded heatsink. The number X denotes the FEM discretization model. Higher X is means larger matrices. In this case, it corresponds to the number of FEM refinements, although no FEM knowledge is needed for this exercise.

For problems (d)-(e), you should not compute explicitly the matrix M^{-1}, but instead only compute its' action. This typically requires a modification of the implementation of GMRES and PCG. You may use the backslash command to compute the action of T_1^{-1} and T_2^{-1}, which is fast since they are upper and lower triangular matrices.



(f) Optional: Repeat the experiment in (c)-(e) for $X = 2$ and $X = 3$.

Below is only for PhD students taking the course *FSF2580 Numerical linear algebra*. The same rules apply as for homework 1, i.e., you may skip one of the SF2524-problems above

7. In the lectures we derived convergence bounds of GMRES for diagonalizable matrices. In this exercise you shall show convergence for a class of non-diagonalizable matrices. Suppose $A \in \mathbb{C}^{m\times m}$ is invertible and suppose $\lambda_1 = \lambda_2$ is a double eigenvalue and all other eigenvalues are distinct. Assume that λ_1 has a Jordan block of size two. Moreover, suppose all eigenvalues λ_i, $i = 1, \dots, m$ are contained in an open disk of radius $\rho > 0$ centered at $c \in \mathbb{C}$, such that $\lambda_i \in C(\rho, c)$ for all $i = 1, \dots, m$. Assume $|\rho| < |c|$ and $\lambda_1 \neq c$.

   (a) Let $V\Lambda V^{-1} = A$ be the Jordan canonical form. Prove

   $$\min_{p\in P_n^0} \|p(A)\| \leq \|V\|\|V^{-1}\| \min_{p\in P_n^0} \|p(\Lambda)\|$$

   (b) Prove that for any polynomial $p(z) = a_0 + a_1 z + \cdots + a_n z^n$,

   $$p\left(\begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix}\right) = \begin{pmatrix} p(\lambda_1) & p'(\lambda_1) \\ 0 & p(\lambda_1) \end{pmatrix}.$$

   (c) Prove

   $$\|p(\Lambda)\| = \max\left(\left\|\begin{pmatrix} p(\lambda_1) & p'(\lambda_1) \\ 0 & p(\lambda_1) \end{pmatrix}\right\|, |p(\lambda_3)|, |p(\lambda_4)|, \dots, |p(\lambda_m)|\right)$$

   (d) Determine α_n and β_n such that

   $$p(z) = (\alpha_n + \beta_n z)\frac{(c-z)^{n-1}}{c^{n-1}}$$

   satisfies $p \in P_n^0$ and $p'(\lambda_1) = 0$ for all $n > 1$.

   (e) Combine (a)-(d) and determine a *bounded* sequence $[\gamma_n]_{n=1}^{\infty}$ such that

   $$\frac{\|Ax_n - b\|}{\|b\|} \leq \|V\|\|V^{-1}\|\gamma_n \frac{\rho^n}{|c|^n}$$

   for all $n > 0$.

   (f) Compare what happens when we carry out many iterations, i.e., when $n \to \infty$? Is there a penalty to have double eigenvalues in the sense of bounds, i.e., is the asymptotic convergence predicted in (e) faster than the prediction we derived for diagonalizable matrices? Point out similarities and differences in the convergence factor, as well as the factor in front of the convergence factor term.

Recall from the definition of the Jordan canonical form: If the eigenvalue λ_1 has one Jordan block of size two and all other Jordan blocks are of size one we have the following factorization. There exists an invertible matrix $V \in \mathbb{C}^{m\times m}$ and a matrix

$$\Lambda = \begin{pmatrix} \lambda_1 & 1 & & & \\ & \lambda_1 & 0 & & \\ & & \lambda_3 & \ddots & \\ & & & \ddots & 0 \\ & & & & \lambda_m \end{pmatrix}.$$

such that $A = V\Lambda V^{-1}$.

Hint for (c): Show that the 2-norm of a block diagonal matrix is the maximum of the two-norm of the blocks by using the formula for the two-norm in terms of singular values.

From lectures: $P_n^0 = \{p \in P_n : p(0) = 1\}$ where P_n is the set of polynomials of degree less or equal to n.

Connection with current research: Researchers in numerical linear algebra are actively working on gaining further understanding of $\|p(A)\|$. The set $W(A) = \{\frac{x^*Ax}{x^*x} : x \in \mathbb{C}^m\}$ is called the field of values of A. It has been shown that $\|p(A)\| \leq \alpha \cdot \max_{z\in W(A)} |p(z)|$ holds for $\alpha = 11.08$. The open problem called Crouzeix's conjecture states that the bound holds for $\alpha = 2$. What happens if A is a Jordan block of size two with $\lambda_1 = \lambda_2 = 0$ and $p(z) = z$? Can you show that α cannot be smaller than 2? *Spoiler alert* See presentation at an important conference: http://sites.uclouvain.be/HHXIX/Plenaries/Overton.pdf

8. Only for PhD students attending SF3580 and have attended SF2524 in their Master studies: GMRES can be adapted to incorporate a starting value (called x_0), which should be selected as a vector approximating the solution to $Ax = b$. We try to find a correction which is an element of a Krylov subspace such that we instead define the approximations as

$$\min_{x \in \mathcal{K}(A,f)+x_0} \|Ax - b\|_2 = \|Ax_n - b\|_2,$$

which reduces to the definition of GMRES-iterates if we select $x_0 = 0$ and $f = b$.

(a) Suppose $AQ_n = Q_{n+1}\underline{H}_n$ is an Arnoldi factorization. How should f be selected such that we can generalize lemma in lecture notes and

$$x_n = Q_n z + f$$

(b) Suppose A is diagonalizable and generalize the min-max bound

(c) Use the above as a procedure to carry out explicit restarting and illustrate it with an example.