

Homework 3

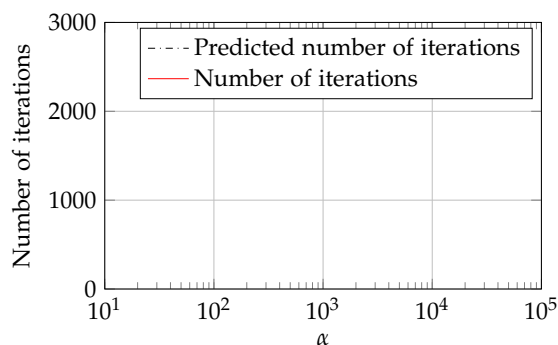
1. Create problems and solutions on the course training wiki before the wiki deadline. This task is optional for SF2524, but can give you more bonus points. See instructions here:

CANVAS → SF2524 → Modules → Homework & Bonus points rules

Note in particular: Please do not solve more problems than you post. SF3580-students are expected to post at least one problem three days before the wiki deadline.

For this homework set you can post or answer wiki problems in Block 3 or Block 4.

2. **Exercise about basic QR-method.** Implement the basic QR-method. Apply it to `alpha_example.m` from the course web page. Measure the error with the maximum value below the diagonal `errfun=@(A) max(max(abs(tril(A,-1))))`.
 - (a) Plot the number of iterations required to achieve error 10^{-10} , as a function of α . More precisely, generate the following plot (with `semilogx()`)



- (b) Suppose the eigenvalues are ordered by magnitude $|\lambda_1| < \dots < |\lambda_m|$. From the lecture notes we know that the elements below the diagonal will asymptotically after n iterations be proportional to $|\lambda_i/\lambda_j|^n$ with $i < j$. For large α the error will be dominated by one particular choice of i and j . Which ones?
- (c) Use (b) to establish an estimated number of iterations required to reach a specified tolerance, for different choices of α . Add a

For the theoretical reasoning in (b) and (c) you may use the function `eig`

Hint for (c): Show that if the error behaves as $e_k = |\beta|^k$, then $e_N = \text{TOL}$ if $N = \ln(\text{TOL}) / \ln(|\beta|)$.

plot of the predicted number of iterations in the plot generated in (a), for tolerance 10^{-10} , and discuss the result.

3. Exercises about Hessenberg reduction and shifted QR-method.

Hint for (a): First derive a formula first for the case $\|y\| = 1$.

- (a) Generalize the lemma about Householder reflectors in the lecture notes as follows. Given a vector $x \in \mathbb{R}^n$ and a vector $y \in \mathbb{R}^n$ with $y \neq 0$ and $x \neq 0$, derive a formula for a Householder reflector (represented by a normal direction $u \in \mathbb{R}^n$) such that $Px = \alpha y$ for some value α .
- (b) Implement a naive (inefficient) Hessenberg reduction by completing the program `naive_hessenberg_red.m` on the course web page.
- (c) Implement Algorithm 2 in the lecture notes and compare the computation time with the algorithm in (b). Carry out the comparison by computing a Hessenberg reduction of $A = \text{alpha_example}(1, m)$, which generates an $m \times m$ -matrix. Complete the following table.

	CPU-time Algorithm 2	CPU-time of algorithm in (b)
m=10		
m=100		
m=200		
m=300		
m=400		

- (d) Let $\bar{H}$ be the result of one step of the shifted QR-method with shift σ for the matrix

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

Run the shifted QR for two different choices of σ and complete the following table

ε	$ \bar{h}_{2,1} $	
	$\sigma = 0$	$\sigma = a_{2,2}$
0.4	0.0961	0.0769
0.1		
0.01		
$\vdots$		
10^{-10}		
0		

Interpret the result in the table. What do the values in the table correspond to? Which choice of σ is better in this case?

Hint: What does the shifted QR-method reduce to when you select $\sigma = 0$?

4. Do the Quiz 3 homework on CANVAS. This task is individual. No hand-in answer necessary.
5. Download the template `schur_parlett.m` from the course web page.

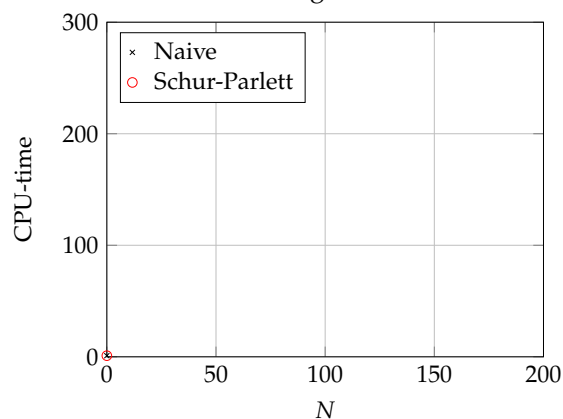
(a) Complete the template code and compute $\sin(A)$ where

$$A = \begin{bmatrix} 1 & 4 & 4 \\ 3 & -1 & 3 \\ -1 & 4 & 4 \end{bmatrix}$$

- (b) Let A be defined by `A=rand(100,100); A=A/norm(A);`. Use the Schur-Parlett method from (a) to compute A^N , and compare with the naive method to compute A^N :

`B=A; for i=1:N-1; B=B*A; end.`

Complete the following figure. Increase N until you see that the best method changes, or see a tendency regarding which method will be better for large N .



The MATLAB command `A^N` for large N will actually not do the naive method. For large N MATLAB switches and uses an underlying procedure similar to Schur-Parlett.

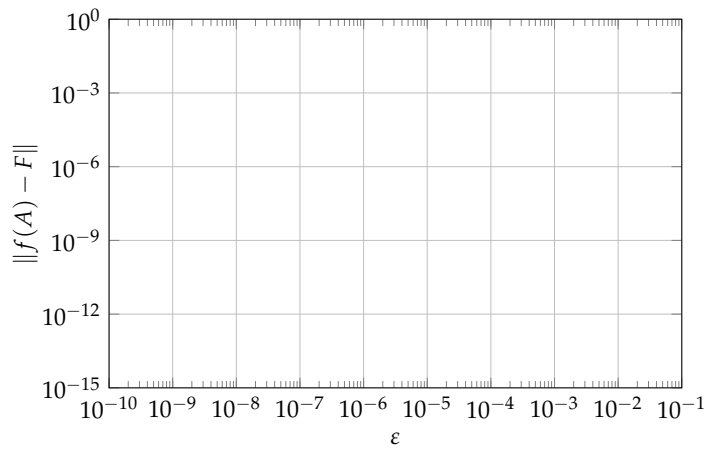
Make appropriate sampling of the x -axis in the figure to support the conclusion, for instance $N = 10, 50, 100, 150, 200, \dots$

- (c) What are the theoretical computational complexities of the two methods in (b) as a function of the size of the matrix n and N ? In other words, find p and q such that the number of flops $\sim \mathcal{O}(n^p N^q)$ for the two methods.

6. Let $A = \begin{bmatrix} \pi & 1 \\ 0 & \pi + \varepsilon \end{bmatrix}$ where $\varepsilon > 0$.

- (a) Prove (for instance using the Jordan definition) that if p is a polynomial which interpolates g in the eigenvalues of A , then $p(A) = g(A)$. Find exact expressions for α and β when $p(z) = \alpha + \beta z$ for the matrix A .
- (b) Give a formula for the exact value of $\exp(A)$ using (a).

- (c) Let F be the result of computing $\exp(A)$ with the Jordan definition as in the last example in Section 3.1.2. Compare the exact result in (b) with the computed value for different ε . Generate the following figure (using `loglog()`) and explain the result.



Only for PhD students taking the course *Numerical linear algebra SF3580*:

7. Exercise about exploitation of structure in specific application.

The purpose of this exercise is to learn some techniques to derive more efficient methods by taking problem-specific structure into account. (The new method you will derive is not necessarily in general the best for this problem-type.)

(a) Prove that

$$\frac{d}{dt} \exp(tA) = A \exp(tA) = \exp(tA)A$$

(b) Let $G(t) := \exp(-tA)B \exp(tA)$ and let $[\cdot, \cdot]$ denote a commutator, i.e., $[A, B] := AB - BA$. Show that

$$G(t) = B + t[B, A] + \frac{t^2}{2!}[[B, A], A] + \frac{t^3}{3!}[[[B, A], A], A] + \dots \quad (*)$$

(c) Suppose A is anti-symmetric $A^T = -A$. Let

$$P := \int_0^\tau \exp(tA^T)B \exp(tA) dt$$

Derive an expression for P involving commutators of A and B .

(d) Let $C_k = [C_{k-1}, A]$ with $C_0 = B$. Show that $\|C_k\| \leq 2^k \|A\|^k \|B\|$.

(e) Suppose $\|A\| < \frac{1}{2}$ and $t \leq 1$. Let G_N be the truncation of G ,

$$G_N(t) := B + t[B, A] + \dots + \frac{t^N}{N!} [\dots [[B, A], A] \dots, A].$$

Derive a bound for $\|G_N(t) - G(t)\|$, which converges to zero as $N \rightarrow \infty$ for any $t \leq 1$.

(f) Combine (c)-(e) and derive a numerical method to compute P where A is anti-symmetric and $\|A\| < 1/2$. Construct the algorithm such that the user can specify a tolerance.

(g) Compare your numerical method with the naive numerical integration approach:

```
P=integral(@(t) expm(t*A')*B*expm(t*A),0,T,'arrayvalued',true);
```

Use $\tau = 1$ and the matrices generated by:

```
A=gallery('neumann',20^2); A=A-A'; A=A/(2*norm(A,1));
B=sprandn(length(A),length(A),0.05);
```

How much better is the new method?

Connection with current research: In the field of quantum chemistry, the relation (*) for $t = 1$ is commonly called the Baker-Campbell-Hausdorff expansion. It is fundamental in one of the leading numerical methods in that field - the so-called coupled cluster approach.

The quantity P is called a Gramian, and it is often used in system and control in order to study controllability, observability and to derive optimal control as well as carrying out "model order reduction".

Not a part of the exercise: Can you derive a similar algorithm which does not require the matrix to be anti-symmetric?

8. will appear in a later version