

SF3580

HW 3

Anna Broms & Fredrik Fryklund

2018/11/29

Task 2

Task 4

Task 5

(a)

The matrix A is diagonalizable with the eigendecomposition $A = QDQ^{-1}$, where D is a diagonal matrix. For such structures it holds that $\sin(A) = Q \sin(D)Q^{-1}$. Thus we can validate the result for the Schur-Parlett method, which is

$$\sin(A) = \sin\left(\begin{bmatrix} 1 & 4 & 4 \\ 3 & -1 & 3 \\ -1 & 4 & 4 \end{bmatrix}\right) \approx \begin{bmatrix} 0.846192 & 0.0655435 & -0.187806 \\ 0.33476 & 0.385017 & -0.141244 \\ -0.190921 & 0.192478 & 0.848269 \end{bmatrix}. \quad (1)$$

which in norm differs $4.28e - 16$ from $Q \sin(D)Q^{-1}$.

(b) & (c)

It is clear from Figure 1 that the number of flops required for Schur-Parlett is not discernibly affected by N , at least for $N \in 10, 50, 100, 150, 200, 250, 300$. This is not surprising, as often the most computationally demanding part of the Schur-Parlett method, is in performing the Schur decomposition, which scales like $O(n^3)$. Once obtained, the function f is only applied to the diagonal elements, which are scalars.

For the naive approach the number of flops is proportional to N . A matrix multiplication is of $O(n^3)$, thus performing N matrix gives $O(Nn^3)$, which we read from Figure 1. The black line corresponds to the line $0.08 + 0.07 N$.

Task 6

(a)

The matrix

$$A = \begin{bmatrix} \pi & 1 \\ 0 & \pi + \varepsilon \end{bmatrix}, \quad (2)$$

with $\varepsilon > 0$, has two eigenvalues: $\lambda_1 = \pi$ and $\lambda_2 = \pi + \varepsilon$. Let $A = X \text{diag}(J_1, J_2) X^{-1}$, be the Jordan canonical form, with $J_1 = \lambda_1$ and $J_2 = \lambda_2$.

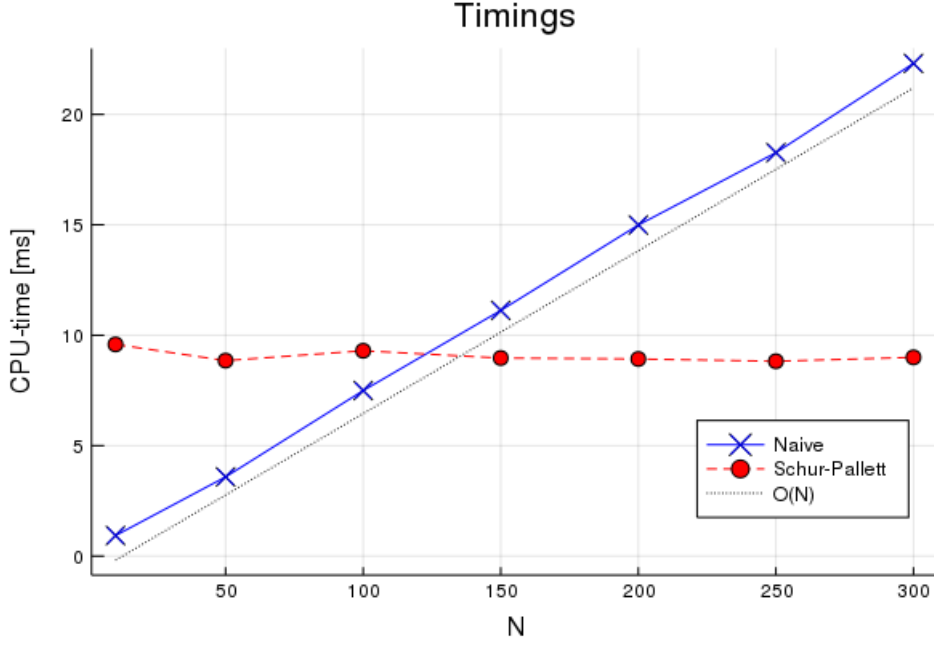


Figure 1: Task 5, (b) & (c): CPU-time in milliseconds, as a function of N .

The Jordan canonical form definition gives that

$$p(A) = X \text{diag}(p(J_1), p(J_2)) X^{-1}. \quad (3)$$

A simple consequence is

$$g(A) = X \text{diag}(g(\lambda_1), g(\lambda_2)) X^{-1} = X \text{diag}(p(\lambda_1), p(\lambda_2)) X^{-1} = p(A), \quad (4)$$

since the polynomial p interpolates the function g in the eigenvalues of A , i.e. $p(\lambda_1) = g(\lambda_1)$ and $p(\lambda_2) = g(\lambda_2)$.

Two points defines a unique polynomial of order 1, thus we may choose a p in \mathbb{P}^1 and write $p(z) = \alpha + \beta z$. The unknown coefficients are obtained by solving

$$\begin{cases} \alpha + \beta \lambda_1 = g(\lambda_1) \\ \alpha + \beta \lambda_2 = g(\lambda_2) \end{cases} \Leftrightarrow \begin{cases} \alpha = \frac{g(\lambda_1)\lambda_2 - g(\lambda_2)\lambda_1}{\lambda_2 - \lambda_1} \\ \beta = \frac{g(\lambda_2) - g(\lambda_1)}{\lambda_2 - \lambda_1} \end{cases} \quad (5)$$

(b)

Given $g := \exp$ we have from (a) and (b) that

$$p(A) = \alpha I + \beta A = \frac{\exp(\pi)(\pi + \varepsilon) - \exp(\pi + \varepsilon)\pi}{\varepsilon} I + \frac{\exp(\pi + \varepsilon) - \exp(\pi)}{\varepsilon} A \quad (6)$$

$$= \frac{\exp(\pi)}{\varepsilon} (\varepsilon I + (1 - \exp(\varepsilon))(\pi I - A)) \quad (7)$$

(c)

It is known that the Jordan decomposition is unstable for non-symmetric matrices, as the eigenvalues may lie close to each other. For the given matrix A this can be tuned artificially by setting $\varepsilon = |\lambda_1 - \lambda_2|$. As we see in Figure (to be added, some layout to fix). However, the function `exp` in Julia is analogous to `expm` in Matlab, (we compared the results for $\varepsilon = 1e-1, \dots, 1e-10$). Using Matlabs `expm(A)` as reference value, the error from the exact result in (b) also increases as ε . This is most likely due to cancellation.

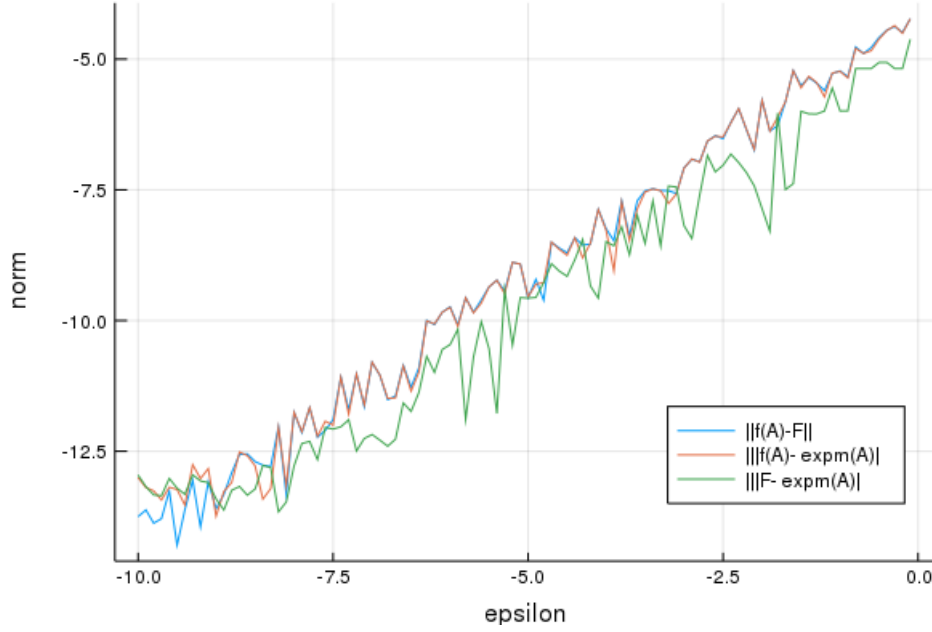


Figure 2: Task 5, (b) & (c): CPU-time in milliseconds, as a function of N .

Task 7

(a)

Consider the function $f(z, t) = e^{tz}$. We want to investigate the matrix valued function $f(A, t) = e^{tA}$. Let $\mu \in \mathbb{C}$ be an expansion point. Then,

$$f(A, t) = \sum_{i=0}^{\infty} \frac{f^{(i)}(\mu)}{i!} (A - \mu I)^i = \sum_{i=0}^{\infty} \frac{t^i e^{t\mu}}{i!} (A - \mu I)^i. \quad (8)$$

If $A \in \mathbb{C}^{n \times n}$, then $f : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$. Now, compute the derivative of $f(A, t)$ with respect to time:

$$\begin{aligned} \frac{d}{dt} e^{tA} &= \frac{d}{dt} \sum_{i=0}^{\infty} \frac{t^i e^{t\mu}}{i!} (A - \mu I)^i = \\ &= \sum_{i=0}^{\infty} \frac{d}{dt} \left(\frac{t^i e^{t\mu}}{i!} (A - \mu I)^i \right) = \\ &= \sum_{i=0}^{\infty} \frac{d}{dt} \left(\frac{t^i e^{t\mu}}{i!} \right) (A - \mu I)^i = \\ &= \sum_{i=0}^{\infty} \left(\frac{it^{i-1} e^{t\mu} + t^i \mu e^{t\mu}}{i!} \right) (A - \mu I)^i. \end{aligned} \quad (9)$$

The last expression can be identified as $g(A)$, where $g(z) = ze^{tz}$ as the expression

$$(it^{i-1} e^{t\mu} + t^i \mu e^{t\mu}) \quad (10)$$

is the i th derivative of the product $z \cdot e^{tz}$, which can be seen using the general Leibniz rule $((f_1 f_2)^{(n)}) = \sum_{k=0}^n \binom{n}{k} f_1^{(n-k)}(x) f_2^{(k)}(x)$. Thus, we can conclude that $\frac{d}{dt} e^{tA} = A e^{tA}$. The matrix function $e^{tA} A$ has the same Taylor expansion expression as $A e^{tA}$. Thus, $\frac{d}{dt} e^{tA} = A e^{tA} = e^{tA} A$, which is what we wanted to show.

(b)

Introduce

$$[B, A]_n = [[B, A]_{n-1}, A], n = 0, 1, 2, \dots, \quad \text{where } [B, A]_1 = [B, A] = BA - AB \text{ and } [B, A]_0 = B, \quad (11)$$

which satisfies $[A + B, C]_n = [A, C]_n + [B, C]_n$. This is shown by induction: the initial case is $[A + B, C]_1 = AC - CA + BC - CB = [A, C]_1 + [B, C]_1$. Now assume $[A + B, C]_n = [A, C]_n + [B, C]_n$ holds, then

$$[A + B, C]_{n+1} = [AC - CA + BC - CB, A]_n = [AC - CA, C]_n + [BC - CB, C]_n \quad (12)$$

$$= [[A, C], C]_n + [[B, C], C]_n = [A, C]_{n+1} + [B, C]_{n+1}. \quad (13)$$

Let $G(t) = \exp(-tA)B\exp(tA)$, which is analytic in t . Thus we may write

$$G(t) = \sum_{i=0}^{\infty} \frac{t^i}{i!} G^{(i)}(\mu) = \sum_{i=0}^{\infty} \frac{t^i}{i!} G_i, \quad (14)$$

where $G_0 = B$. By (a) we have that

$$\frac{d}{dt} G(t) = G(t)A - AG(t) = [G(t), A]. \quad (15)$$

Setting this to be equal to the the derivative of (14) with respect to t gives

$$\sum_{i=0}^{\infty} \frac{t^i}{i!} [G_i, A] = \sum_{i=1}^{\infty} \frac{t^{i-1}}{(i-1)!} G_i. \quad (16)$$

By shifting the indexing from $i = 1, 2, \dots$ to $i = 0, 1, \dots$ for the right hand side we get

$$\sum_{i=0}^{\infty} \frac{t^i}{i!} [G_i, A] = \sum_{i=0}^{\infty} \frac{t^i}{i!} G_{i+1}. \quad (17)$$

We conclude that $G_{i+1} = [G_i, A]_i$, that is $G_1 = [G_0, A]_0 = B$ and

$$G(t) = B + t[B, A] + \frac{t^2}{2!} [[B, A], A] + \frac{t^2}{2!} [[B, A], A, A] + \dots \quad (18)$$

(d)

Task: Let $C_k = [C_{k-1}, A]$, with $C_0 = B$. We want to show that $\|C_k\| \leq 2^k \|A\|^k \|B\|$.

The proof is done by induction. For $k = 0$ we have that $\|C_0\| = \|B\| \leq 2^0 \|A\|^0 \|B\|$. Now, assume that $\|C_k\| \leq 2^k \|A\|^k \|B\|$. We want to show that $\|C_{k+1}\| \leq 2^{k+1} \|A\|^{k+1} \|B\|$:

$$\begin{aligned} \|C_{k+1}\| &= \|C_k A - A C_k\| = \|C_k A + (-A C_k)\| \leq \|C_k A\| + \|-A C_k\| = \|C_k A\| + \| -1 \| \|A C_k\| \leq \|C_k\| \|A\| + \|A\| \|C_k\| = \\ &= 2 \|A\| \|C_k\| = 2^{k+1} \|A\|^{k+1} \|B\|, \end{aligned} \quad (19)$$

which is what we wanted to show.

(e)

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

$$G(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} C_k. \quad (20)$$

Then,

$$\begin{aligned}
\|G_N(t) - G(t)\| &= \left\| \sum_{k=N+1}^{\infty} \frac{t^k}{k!} C_k \right\| \leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!} \right) \|C_k\| \leq \\
&\leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!} \right) 2^k \|A\|^k \|B\| \leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!} \right) 2^k \left(\frac{1}{2} \right)^2 \|B\| \leq \frac{\|B\|}{(N+1)!} \sum_{k=N+1}^{\infty} t^k = \frac{\|B\|}{(N+1)!} \sum_{k=0}^{\infty} t^{k+(N+1)} = \\
&= \frac{\|B\| t^{N+1}}{(N+1)!} \cdot \frac{1}{1-t}.
\end{aligned} \tag{21}$$