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

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

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 leat for $N \in 10, 50, 100, 150, 200, 250, 300$. This is not suprsining, as often the most computationally demanding part of the Schur–Parlett method, in is 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 appraoch 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

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

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

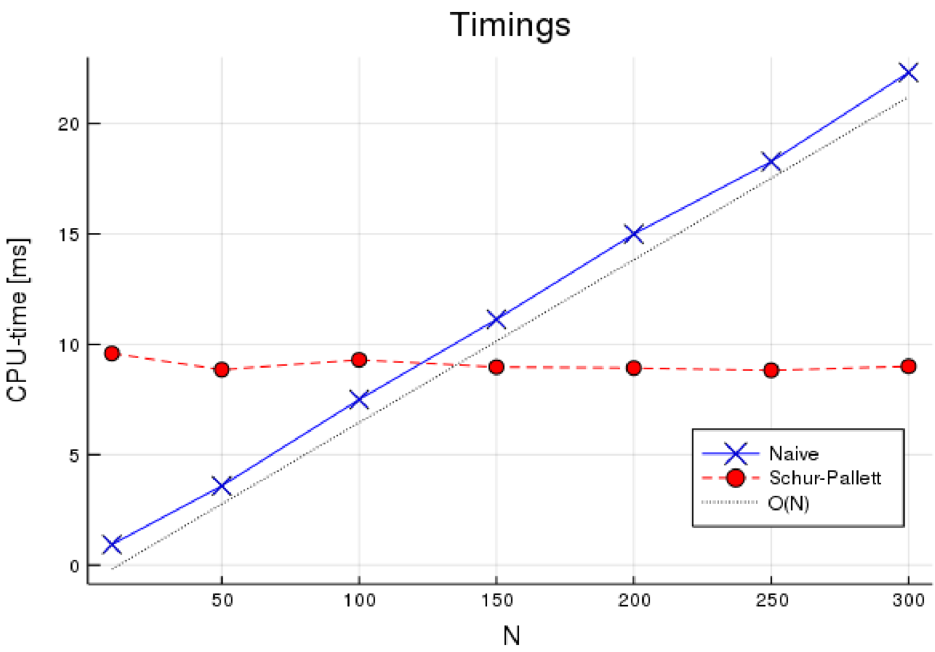

Figure 1: Task 5, (b) & (c): CPU–time in milliseconds, as a function of N.

The Jordan canonical form definition gives that

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

A simple consequence is

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

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 coefficents are obtianed by solving

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

(b)

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

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

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

(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 resluts for $\varepsilon = 1e-1, \ldots, 1e-10$). 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.

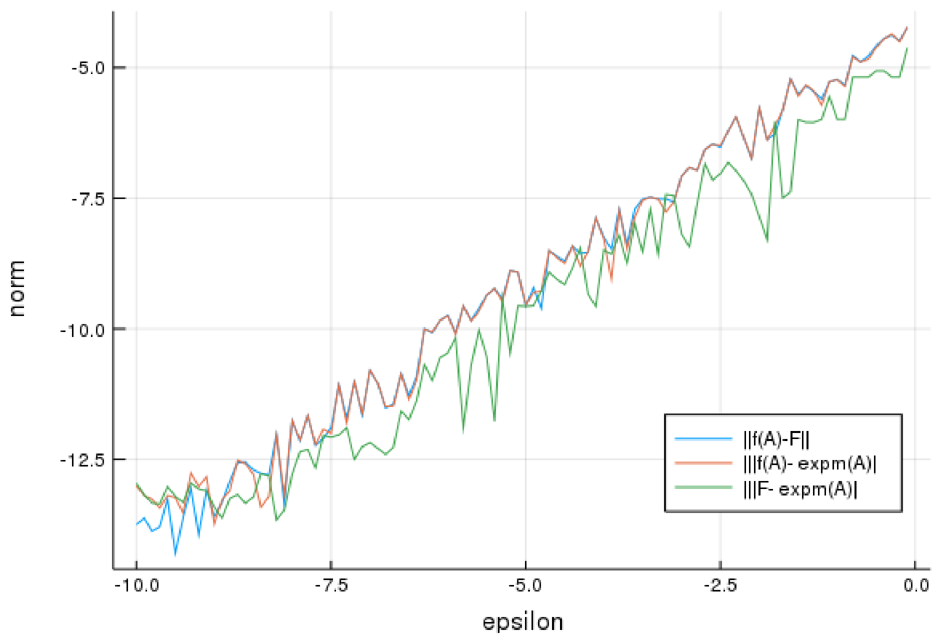

Figure 2: Task 5, (b) & (c): CPU–time in milliseconds, as a function of N.

Task 7

(a)

Comment: We have two alternative solutions in this excersise where we have used the definition of a matrix function in two different ways. Please comment if the second way to solve the problem also is valid, where we see the function f as a function of two variables $f = f(z, t)$ and make a Taylor expansion in z only.

Solution

Let $\mu \in \mathbb{C}$ be an expansion point, then

$$f(tA) = \sum_{i=0}^{\infty} \frac{f^{(i)}(\mu)}{i!}(tA - \mu I)^i. \tag{8}$$

For $f(z) = \exp(z)$ it is analytic and we can without loss of generality set $\mu = 0$. For now assume that $\frac{d}{dt}(tA)^i = iA(tA)^{i-1}$, then

$$\frac{d}{dt}\exp(tA) = \frac{d}{dt}\sum_{i=0}^{\infty}\frac{\exp(0)}{i!}(tA - 0I)^i = \sum_{i=0}^{\infty}\frac{1}{i!}\frac{d}{dt}(tA)^i = A\sum_{i=1}^{\infty}\frac{1}{(i-1)!}(tA)^{i-1} = A\sum_{i=0}^{\infty}\frac{1}{i!}(tA)^i = A\exp(tA), \tag{9}$$

by shifting the indices $i \to i + 1$. We now motivate the claim above. By definition

$$\frac{d}{dt}(tA)^i = \lim_{\epsilon\to 0}\frac{((t+\varepsilon)A)^i - (tA)^i}{\varepsilon} = \lim_{\epsilon\to 0}\frac{(t+\varepsilon)^i - t^i}{\varepsilon}A^i = \frac{d}{dt}(t^i)A^i = iA(tA)^{i-1}. \tag{10}$$

Since $it^{i-1}A^{i-1}A = iAt^{i-1}A^{i-1}$ we have $A\exp(tA) = \exp(tA)A$.

Alternative solution

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

$$f(A,t) = \sum_{i=0}^{\infty} \frac{f^{(i)}(\mu,t)}{i!}(A-\mu I)^i = \sum_{i=0}^{\infty} \frac{t^i e^{t\mu}}{i!}(A-\mu I)^i. \tag{11}$$

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

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}e^{tA} &= \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{\infty}\frac{t^i e^{t\mu}}{i!}(A-\mu I)^i = \\ &= \sum_{i=0}^{\infty}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{t^i e^{t\mu}}{i!}(A-\mu I)^i\right) = \\ &= \sum_{i=0}^{\infty}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{t^i e^{t\mu}}{i!}\right)(A-\mu I)^i = \\ &= \sum_{i=0}^{\infty}\left(\frac{it^{i-1}e^{t\mu} + t^i\mu e^{t\mu}}{i!}\right)(A-\mu I)^i.\end{aligned} \tag{12}$$

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

$$\left(it^{i-1}e^{t\mu} + t^i\mu e^{t\mu}\right) \tag{13}$$

is the ith 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 $\frac{\mathrm{d}}{\mathrm{d}t}e^{tA} = Ae^{tA}$. The matrix function $e^{tA}A$ has the same Taylor expansion expression as Ae^{tA}. Thus, $\frac{\mathrm{d}}{\mathrm{d}t}e^{tA} = Ae^{tA} = e^{tA}A$, which is what we wanted to show.

(b)

Introduce

$$[B,A]_n = [[B,A]_{n-1},A], n = 0,1,2,\dots, \quad \text{where } [B,A]_1 = [B,A] = BA - AB \text{ and } [B,A]_0 = B, \tag{14}$$

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

$$[A+B,C]_{n+1} = [AC - CA + BC - CB, A]_n = [AC - CA, C]_n + [BC - CB, C]_n \tag{15}$$

$$= [[A,C],C]_n + [[B,C],C]_n = [A,C]_{n+1} + [B,C]_{n+1}. \tag{16}$$

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

$$G(t) = \sum_{i=0}^{\infty}\frac{t^i}{i!}G^{(i)}(\mu) = \sum_{i=0}^{\infty}\frac{t^i}{i!}G_i, \tag{17}$$

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

$$\frac{d}{dt}G(t) = G(t)A - AG(T) = [G(t),A]. \tag{18}$$

Setting this to be equal to the the derivative of (17) with respect to t gives

$$\sum_{i=0}^{\infty}\frac{t^i}{i!}[G_i,A] = \sum_{i=1}^{\infty}\frac{t^{i-1}}{(i-1)!}G_i. \tag{19}$$

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

$$\sum_{i=0}^{\infty} \frac{t^i}{i!}[G_i, A] = \sum_{i=0}^{\infty} \frac{t^i}{i!} G_{i+1}. \tag{20}$$

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

$$G(t) = B + t[B, A] + \frac{t^2}{2!}[[B, A], A] + \frac{t^2}{2!}[[B, A], A, A] + \ldots \tag{21}$$

(c)

We identify the integrand as $G(t)$, that is

$$P = \int_0^{\tau} \exp(tA^T) B \exp(tA)\, dt = \int_0^{\tau} \exp(-tA) B \exp(tA)\, dt = \int_0^{\tau} G(t)\, dt. \tag{22}$$

Introduce

$$P_n = \int_0^{\tau} \sum_{i=0}^{n} \frac{t^i}{i!} G_{i+1}\, dt = \sum_{i=0}^{n} \int_0^{\tau} \frac{t^i}{i!} G_{i+1}\, dt, \tag{23}$$

Since the integrand is uniformly convergent, assuming τ finite, it holds that

$$\lim_{n\to\infty} P_n = \lim_{n\to\infty} \int_0^{\tau} \sum_{i=0}^{n} \frac{t^i}{i!} G_{i+1}\, dt = \int_0^{\tau} \lim_{n\to\infty} \sum_{i=0}^{n} \frac{t^i}{i!} G_{i+1}\, dt = \int_0^{\tau} G(t)\, dt = P \tag{24}$$

where the limit was moved inside due to the dominated convergence theorem. Furthermore,

$$\int_0^{\tau} \frac{t^i}{i!} G_{i+1}\, dt = [G_i, A] \frac{\tau^{i+1}}{(i+1)!}, \tag{25}$$

for every i. Thus

$$P = \lim_{n\to\infty} \sum_{i=0}^{n} \int_0^{\tau} \frac{t^i}{i!} G_{i+1}\, dt = \sum_{i=0}^{\infty} [G_i, A] \frac{\tau^{i+1}}{(i+1)!}. \tag{26}$$

(d)

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

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

$$\begin{aligned} ||C_{k+1}|| = ||C_k A - A C_k|| = ||C_k A + (-A C_k)|| \le ||C_k A|| + ||-A C_k|| = ||C_k A + |-1|||A C_k|| \le ||C_k|| ||A|| + ||A|| ||C_k|| = \\ = 2||A|| ||C_k|| = 2^{k+1} ||A||^{k+1} ||B||, \end{aligned} \tag{27}$$

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

$$G(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} C_k. \tag{28}$$

Then,

$$\begin{aligned} \|G_N(t) - G(t)\| &= \| \sum_{k=N+1}^{\infty} \frac{t^k}{k!} C_k \| \leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!}\right) \|C_k\| \leq \\ &\leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!}\right) 2^k \|A\|^k \|B\| \leq \sum_{k=N+1}^{\infty} \left(\frac{t^k}{k!}\right) 2^k \left(\frac{1}{2}\right)^2 \|B\| \leq \frac{\|B\|}{(N+1)!} \sum_{k=N+1}^{\infty} t^k = \frac{\|B\|}{(N+1)!} \sum_{k=0}^{\infty} t^{k+(N+1)} = \\ &= \frac{\|B\|}{(N+1)!} t^{N+1} \cdot \frac{1}{1-t}, \end{aligned} \tag{29}$$

where we have first used the result from (d) and then computed the resulting geometric series.

(f)

Using the results from (c), with

$$P = \sum_{i=0}^{\infty} G_{i+1} \frac{\tau^{i+1}}{(i+1)!} = \sum_{i=1}^{\infty} G_i \frac{\tau^i}{i!} = \sum_{i=0}^{\infty} G_i \frac{\tau^i}{i!} - B, \tag{30}$$

we denote by P^N the computation of P truncated at N terms and obtain that

$$\|P - P^N\| = \|G_N(\tau) - G(\tau)\|. \tag{31}$$

Using the estimate in (e), we thus have that

$$\|P_N - P\| \leq \frac{\|B\| \tau^{N+1}}{(N+1)!(1-\tau)}. \tag{32}$$

Specifying a tolerance in a numerical algorithm for computing P, this last expression can be used to determine the number of iterations needed to compute P given a specified tolerence. We can now design an algorithm for P (in pseudocode) as:

G = B; - Sets initial element G_0.
P = 0; - Sets sum to zero
i = 1; - Number of iterations
err = 1;
t = 1;
while *err >tol* **do**
  G = GA-AG;
  t = tτ/i;
  P = P+Gt;
  i = i+1;
  err = $\frac{\|B\|\tau^{i+1}}{(i+1)!(1-\tau)}$
end

Algorithm 1: Algorithm for computing the integral P.