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 $J_2 = \lambda_2$.

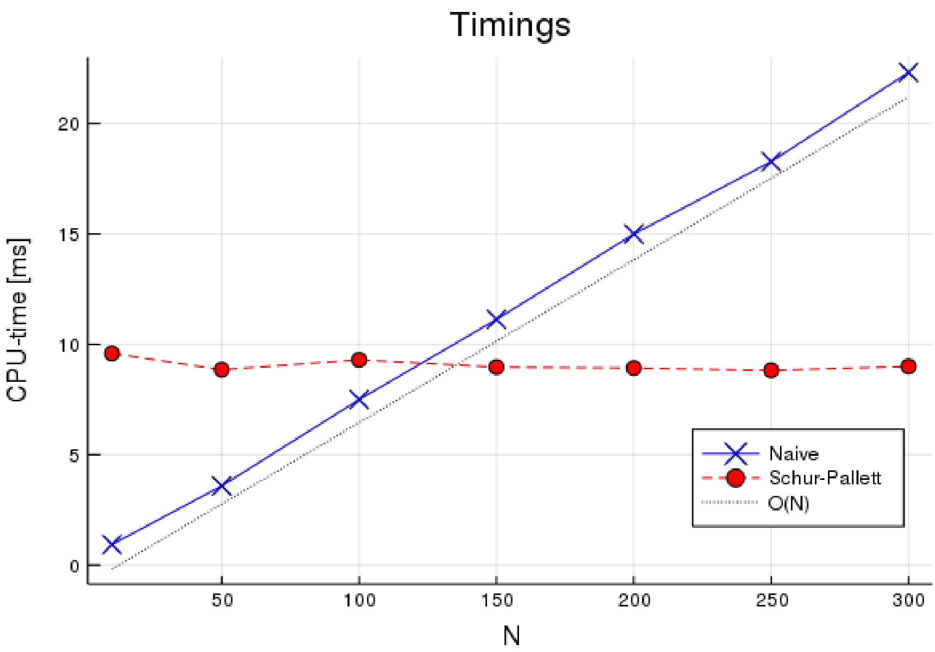

Figure 1: Task 5, (b) & (c): CPU–time in milliseconds, as a function of N.

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.

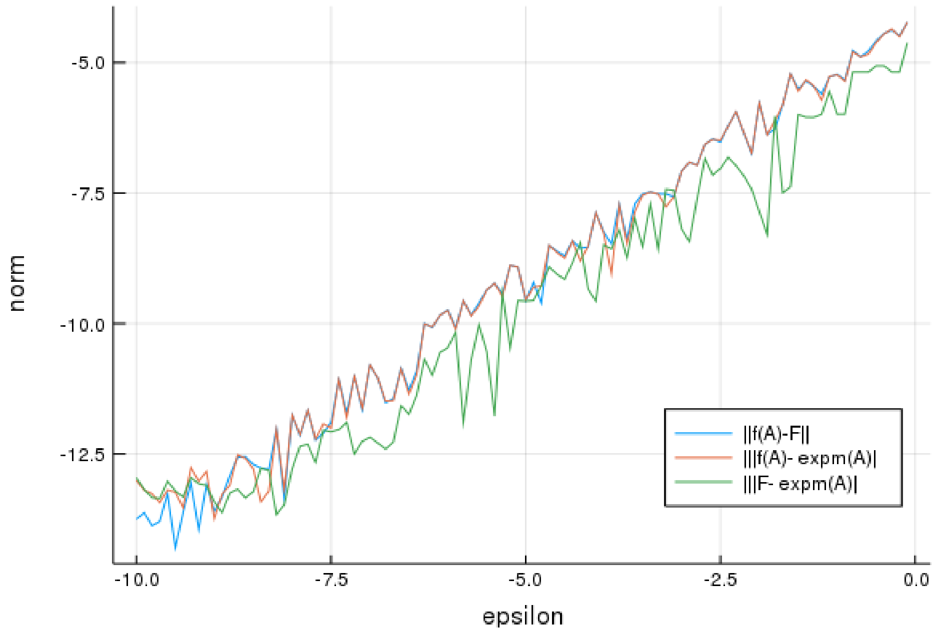

Figure 2: Task 5, (b) & (c): CPU–time in milliseconds, as a function of N.

Task 7

(a)

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,

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

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:

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

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

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

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.

(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 $||C_{k+1}|| \leq 2^{k+1} ||A||^{k+1} ||B||$:

$$\begin{aligned}||C_{k+1}|| = ||C_k A - AC_k|| = ||C_k A + (-AC_k)|| \leq ||C_k A|| + || - AC_k|| = ||C_k A + | - 1|||AC_k|| \leq ||C_k|| ||A|| + ||A|| ||C_k|| = \\ = 2||A|| ||C_k|| = 2^{k+1} ||A||^{k+1} ||B||,\end{aligned} \tag{11}$$

which is what we wanted to show.

(e)