Note Title 2023-03-16

Taylor series methods for matrix functions

1. Choose $\alpha = \dfrac{\mathrm{Tr}(A)}{n}$

2. Evaluate $f(A) \approx \sum_{k=0}^{d} \frac{1}{k!} f^{(k)}(\alpha)(A-\alpha I)^k$

Ex in which the method works poorly:

$$A = \begin{bmatrix} 0 & 30 \\ -30 & 0 \end{bmatrix} \qquad \Lambda(A) = \{\pm 30i\} \qquad f(x) = \exp(x)$$

$$\exp(A) = \begin{bmatrix} \cos(30) & \sin(30) \\ -\sin(30) & \cos(30) \end{bmatrix}$$

$$A = 30\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad A^2 = 30^2\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = 30^2\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$A^3 = 30^3\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad A^4 = 30^4\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\exp(A) = \sum_{k\geq 0} \frac{1}{k!} A^k = \begin{bmatrix} (\text{series for } \cos(30)) & (\text{series for } \sin(30)) \\ (\text{series for } \sin(30)) & (\text{series for } \cos(30)) \end{bmatrix}$$

Using Taylor series: $\alpha = 0$.

Problem: the terms with $k \approx 30$ have size $\left\|\frac{A^k}{k!}\right\| \approx 10^{12}$

When summing them (with alternating signs) to produce $\exp(A)$, there is major cancellation.

Recall: when computing $\exp(-30)$ with the Taylor series, there is a similar phenomenon that can be avoided

with the alternative formula $\exp(-30) = \frac{1}{\exp(30)} = \frac{1}{\sum_{k\geq 0} \frac{1}{k!} 30^k}$.

However, with matrices these tricks will not work:

$\exp(-A)$ still has alternating signs, as $-A = A^T$

When we work with matrices we compute with all eigenvalues $\lambda \in \Lambda(A)$ at the same time, and the same Taylor series will not converge quickly and without cancellation on all of them.

Problems with Taylor series:

- slow convergence if $\Lambda(A)$ are very far apart
- this causes accuracy / cancellation problems, but also an increase in computational cost:
  evaluating $p(A) = \sum_{k=0}^{d} c_k A^k$ on $A \in \mathbb{C}^{n\times n}$
  costs $\mathcal{O}(n^3 d)$ with both direct summation and the Horner method.

(Note: there are better methods for evaluating $p(A)$ with complexity $\mathcal{O}(n^3 \sqrt{d})$)

Schur-Parlett method:

Tries to combine factorizations + Taylor series.

Idea: work with the Schur form $A = QUQ^*$

Q orthogonal, U upper triangular

$$f(A) = f(QUQ^*) = Q \underbrace{f(U)}_{\text{f of triangular matrix}} Q^*$$

How to compute $f(U)$ for triangular U?

EX: $U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$ $\quad f(U) = S = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}$

$s_{11} = f(u_{11})$, $s_{22} = f(u_{22})$ simple to compute.

To compute s_{12}, idea: obtaining a formula that involves s_{12} from the relation

$$Uf(U) = f(U)U.$$

(Indeed, $f(U)$ commutes with U; we can consider f to be a polynomial)

$$\begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}\begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

Entry $(1,2)$: $u_{11}s_{12} + u_{12}s_{22} = s_{11}u_{12} + s_{12}u_{22}$

$$(u_{11} - u_{22})s_{12} = s_{11}u_{12} - u_{12}s_{22}$$

$$s_{12} = u_{12}\frac{s_{11} - s_{22}}{u_{11} - u_{22}} = \frac{f(u_{11}) - f(u_{22})}{u_{11} - u_{22}}u_{12}$$

If $u_{11} \neq u_{22}$, I can solve to compute s_{12}.

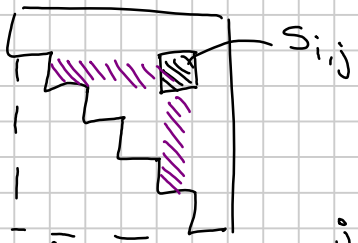
Even for larger matrices, we can set up a similar recurrence:

$$i \to \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & u_{n-1,n} \\ 0 & & & u_{nn} \end{bmatrix}\begin{bmatrix} s_{11} & * & \cdots & * \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ 0 & & & s_{nn} \end{bmatrix} = \begin{bmatrix} s_{11} & * & \cdots & s_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ 0 & & & s_{nn} \end{bmatrix}\begin{bmatrix} u_{11} & * & \cdots & u_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ 0 & & & u_{nn} \end{bmatrix}$$

(i,j) entry (with $j>i$):

$$u_{ii}s_{ij} + u_{i,i+1}s_{i+1,j} + \dots + u_{ij}s_{jj} = s_{ii}u_{ij} + s_{i,i+1}u_{i+1,j} + \dots + s_{ij}u_{jj}$$

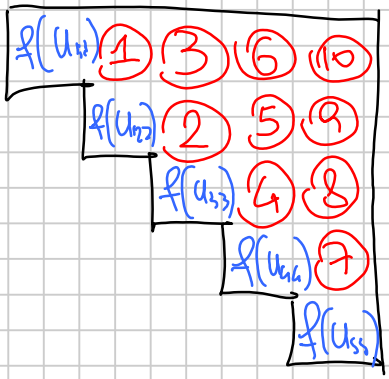
I can solve for s_{ij}, if I know s_{kj} for $k>i$
s_{ik} for $k<j$



i.e., all entries below and on the left

$$s_{ij} = \frac{\sum_{k=i}^{j-1} s_{ik}u_{kj} - \sum_{k=i+1}^{j} u_{ik}s_{kj}}{u_{ii} - u_{jj}}$$

We can compute the entries of the unknown $S=f(U)$ in a suitable order, for instance column by column:



One can compute entries in the order given by the numbers in red.

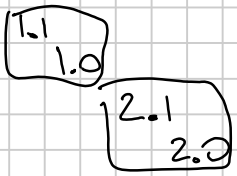
We can carry on the computations if there are no zero denominators, i.e. $\Lambda(A)$ contains no repeated eigenvalues.

$$\begin{bmatrix} s_{i,i} & s_{i,i+1} & \dots & s_{i,j-1} \end{bmatrix} = \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ \vdots \\ u_{j-1,j} \end{bmatrix}$$

```
function Y = funm_parlett(f, A)
[Q, U] = schur(A, 'complex');
% compute S = f(U)
n = size(A,1);
S = zeros(n, n);
for j = 1:n
    S(j,j) = f(U(j,j));
    for i = j-1:-1:1
        S(i,j) = (S(i, i:j-1) * U(i:j-1, j) - U(i, i+1:j) * S(i+1:j, j)) / (U(i,i) - U(j,j));
    end
end
Y = Q*S*Q';
```

Problem: matrices with close eigenvalues have small denominators which cause large errors in the result.

Solution: Idea: apply the recurrence blockwise,



and make sure the blocks contain well-separated eigenvalues

$$U_{ii}S_{ij} + U_{i,i+1}S_{i+1,j} + \dots + U_{ij}S_{jj} = S_{ii}U_{ij} + S_{i,i+1}U_{i+1,j} + \dots + S_{ij}U_{jj}$$

If these are blocks, S_{ij} is the unknown, it is a Sylvester equation in each block.

(with coefficients U_{ii}, U_{jj} already upper triangular)

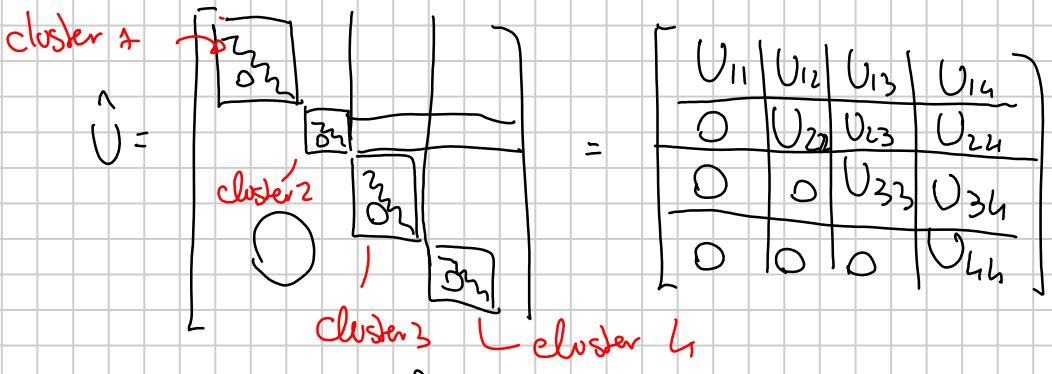
Algorithm (Schur-Parlett method)

1) Compute $A = Q U Q^*$, Schur form

2) Divide the eigenvalues $(t_{11}, t_{22}, \dots t_{nn})$ into clusters, with the goal that eigenvalues in the same cluster are close, and eigenvalues in different clusters are far.

3) Reorder the Schur form so that eigenvalues in the same cluster appear consecutively

$$A = \hat{Q}\hat{U}\hat{Q}^*$$



and subdivide $\hat{U}$ into blocks corresponding to the clusters

$$f(\hat{U}) = S = \begin{bmatrix} S_{11} & \cdots & S_{14} \\ & S_{22} & \vdots \\ & & S_{33} \\ & & & S_{44} \end{bmatrix}$$

4) Compute $f(S_{ii})$ for all diagonal blocks with a Taylor expansion centered in $\frac{1}{\dim} \mathrm{Tr}(S_{ii})$

5) Compute blocks S_{ij} with $j > i$ by solving a Sylvester equation on each block

6) $f(A) = \hat{Q} S \hat{Q}^{\times}$

Issues:

1. Must know how to compute derivatives. Matlab's funm "cheats" and has a few hardcoded functions

2. It is difficult to find the best thing to do when the eigenvalues are <u>not</u> naturally divided in clusters, e.g. equally spaced.

3. The worst case is still one large Taylor expansion, with complexity larger than $O(n^3)$

4. For non-normal matrices, what matters is not $|\lambda_i - \lambda_j|$, but $\text{sep}(U_{ii}, U_{jj})$, more complicated to compute, so it might be less stable than expected.