Homework: Matrix functions

July 12, 2023

This is an assignment for the course 'Numerical Linear Algebra'. The goal of this assignment is to explore matrix functions and associated Krylov methods.

Please note that the questions are deliberately open-ended! Unlike in the exercise sessions, you are encouraged to make your own analysis, provide your own results and construct your own report in your own way. The assignment is individual. Don't be discouraged if not everything works in the end, but show your work and report what you tried and whether you know why something did not work.

Your report should be a standalone text, should read pleasantly and should not refer to this assignment. Take special care of structure and visual presentation. You are allowed to copy text from this assignment document.

1 Theory

In this section we will be looking at how matrix functions can naturally be defined and equivalences between definitions.

1.1 Matrix-valued polynomials, rational functions and power series

Say we have a polynomial $p(t) := \sum_{k=0}^{n} c_k t^k$. Then for any matrix $A \in \mathbb{C}^{n \times n}$ it is natural to define $p(A) := \sum_{k=0}^{n} c_k A^k$, where we adopt the convention that $A^0 = I$. Similarly, for any rational function $f(t) := \frac{p(t)}{q(t)}$ we want to define $f(A) := q(A)^{-1} p(A)$. **When is this undefined? Investigate** $q(A)^{-1}p(A)$

vs. $p(A)q(A)^{-1}$. What do you conclude? Prove this. How do you evaluate $f(A)$ whenever f is given by a power series or a (finite-order) Laurent series?

1.2 Spectrum-based definition

Supposing the matrix A is diagonalizable, i.e. $A = P\Lambda P^{-1}$, there is an obvious definition for $f(A)$, for any f. **Give this definition. Motivate it in the case f is a polynomial, rational function or given by a power series. How restrictive is the assumption that A is diagonalizable? What important matrices are included? Which aren't?** We will mention here that it is possible to extend the above to using the *Jordan canonical form*, but you do not have to do that here.[1]

1.3 Interpolation-based definition

Shockingly, in this section you will show that for any $A \in \mathbb{C}^{n\times n}$ and any sufficiently differentiable f, we can find a polynomial p such that $f(A) = p(A)$. It all starts by observing that only the values that are actually important for matrix *polynomials* are precisely in the spectum of A. First we briefly recall that every matrix $A \in \mathbb{C}^{n\times n}$ with spectrum $\{\lambda_1, \dots, \lambda_k\}$ has a *minimal polynomial* ϕ_A given by

$$\phi_A(t) := \prod_{i=1}^{k}(t-\lambda_k)^{n_k} \tag{1}$$

which is the unique monic minimal degree ($\mathrm{degr}(\phi_A) = n_1 + \cdots + n_k \leq n$) polynomial such that $\phi_A(A) = 0$. We can use it to show theorem 1.

Theorem 1 *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with eigenvalues $\{\lambda_1, \dots, \lambda_k\}$, and minimal polynomial given by equation 1. Then for any two polynomials p_1, p_2 we have that $p_1(A) = p_2(A)$ if and only if*

$$\forall j \in \{1, \dots, k\} : \forall i \in 0, \dots, n_k - 1 : p_1^{(i)}(\lambda_j) = p_2^{(i)}(\lambda_j).$$

Prove this. Hint: if a polynomial p satisfies $p(A) = 0$ then the minimal polynomial of A divides p. To what easier statement does theorem 1 reduce in case A has simple spectrum? Use theorem 1 to motivate the definition 1 of a matrix function.

[1] Though you of course can if you want to.

Definition 1 *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with minimal polynomial given as in equation 1, and let f be a function that is at least $\max_k\{n_k - 1\}$ times differentiable. Say p is its $(n_1, \ldots, n_k)$-*Hermite interpolant *i.e. the polynomial satisfying*

$$\forall j \in \{1, \ldots, k\} : \forall i \in \{0, \ldots, n_k - 1\} : p^{(i)}(\lambda_j) = f^{(i)}(\lambda_j)$$

of minimal degree. Then we define $f(A) = p(A)$.

Comment on this definition. Does it match with the earlier definitions? Is it more general? Is it useful, computable,...?

1.4 The matrix-vector product $f(A)\mathbf{b}$

Have a look at the matrix in the file `Matrix1` provided in the assignment. Call this matrix A. **Study its structure. Calculate exp(A). Study this matrix too. What do you notice? Use this to motivate the following claim:**

Effort should be made to avoid the explicit computation of $f(A)$, if only $f(A)\mathbf{b}$ is needed.

Can you give some examples from the course where only (a small amount of) matrix-vector products are needed?

In this section we will work towards a way to evaluate $f(A)\mathbf{b}$ in an intuitive way.

Firstly, we will define the notion of $\phi_{\mathrm{A},\mathbf{b}}$, i.e. the minimal polynomial of A **with respect to the vector b**. This is simply the polynomial

$$\phi_{\mathrm{A},\mathbf{b}}(t) := \prod_{i=1}^{k} (t - \lambda_i)^{m_k} \tag{2}$$

of minimal degree such that $\phi_{\mathrm{A},\mathbf{b}}(A)\mathbf{b} = \mathbf{0}$. Here $\lambda_1, \ldots, \lambda_k$ are again the eigenvalues of A. **Give a non-trivial example of a pair $(A, \mathbf{b})$ such that $\phi_{\mathrm{A},\mathbf{b}} \neq \phi_A$. Also give one where $\phi_{\mathrm{A},\mathbf{b}} = \phi_A$.** We are now ready to prove theorem 2

Theorem 2 *Let f be a sufficiently differentiable function that has no singularities on the spectrum of a given matrix $A \in \mathbb{C}^{n\times n}$. Then, with p the unique Hermite interpolating polynomial of A w.r.t.* **b** *i.e.*

$$\forall j \in \{1, \ldots, k\} : \forall i \in \{0, \ldots, m_k - 1\} : p^{(i)}(\lambda_j) = f^{(i)}(\lambda_j)$$

we have that $f(A)\mathbf{b} = p(A)\mathbf{b}$.

Prove this. Review the Arnoldi method and also show that theorem 2 implies that

$$f(A)\mathbf{b} = \|\mathbf{b}\|_2 Q_m H_m \mathbf{e}_1$$

where $m := m_1 + \ldots + m_k = \mathbf{degr}(\phi_{\mathrm{A},\mathbf{b}})$**,** $\mathbf{e}_1$ **is the first unit vector in** $\mathbb{R}^n$ **and** Q_m, H_m **are respectively the orthogonal matrix and the Hessenberg matrix of the Arnoldi process after** m **iterations.**

2 Algorithms

In this section we will outline the algorithms you are expected to implement. We distinguish two cases: the 'dense case', where $f(A)$ is to be computed, and the case only $f(A)\mathbf{b}$ is needed for some $\mathbf{b}$.

2.1 The dense case

All the ingredients needed for the computation of $f(A)$ have now been outlined. You have been provided a function `hess_and_phi` that computes the Hessenberg reduction of a given matrix and its associated minimal polynomial. **Study this routine. Briefly explain what it does. What is the format of input and output? How do you evaluate the output?** The reason we chose to do it like this is because `Matlab`'s built-in `minpoly` does not give accurate results in feasable time.[2] You have also been provided a `Matlab` routine called '`hermite`' that computes the Hermite interpolant of a given minimal polynomial, using Newton divided differences. **You do not need to discuss this routine, just understand how to use it.**
Write a `Matlab` routine called `matrix_function` that takes as inputs a function `f`, and a matrix `A` and outputs $f(A)$**, as defined by definition 1. You can use `hess_and_phi` and `hermite`. Test your routine on the matrices in '`test1.mat`' using the script '`test1.m`'.**

[2]More involved techniques for this problem exist, but this is outside of the scope of this assignment.

2.2 $f(A)\mathbf{b}$

In this section you are asked to implement a routine that *approximates* $f(A)\mathbf{b}$. The way this is done is by truncating the Anoldi process at some predefined dimension k, thus resulting in an approximation of $f(A)\mathbf{b}$ in the Krylov space $\mathcal{K}_k(A, \mathbf{b})$. **Write a routine called '`fAb`' that outputs an approximation of $f(A)\mathbf{b}$ and takes as input a function f, a matrix $A \in \mathbf{C}^{n\times n}$, a vector $\mathbf{b} \in \mathbb{C}^n$ and an integer k, representing the maximal dimension of the resulting Krylov space. You can use Arnoldi with iterated Gram-Schmidt. Investigate the convergence of your method for the functions and matrix provided in `test2.mat`. Discuss your findings. Compare the computational efficiency of this method as opposed to explicitely forming $f(A)$. You do not have to use your earlier implementation based on the minimal polynomial; rather than using the theoretical approach above, you can evaluate $f(H_m)$ directly in Matlab. This will result in better comparisons.**

3 Applications

In this section two typical applications of matrix functions are discussed and (simple) model problems are provided for you to solve. It should be stressed that real-world problems quickly become too difficult for these easy standard methods to solve and thus more complex routines are needed. This however falls beyond the scope of this assignment.

3.1 The matrix exponential

Consider the simple system of ODEs

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x},$$

with initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$. Then we know the solution to be given by $\mathbf{x}(t) = e^{At}\mathbf{x}_0$. However, for all but the stablest systems, this is not a good method, due to issues such as stability and stiffness[3]
Here we consider for instance the 2D convection-diffusion equation for the

[3]we will not investigate the issues of stiffness here.

flow $\mathbf{u}(x,y)$:

$$\frac{d\mathbf{u}}{dt} = \epsilon\Delta\mathbf{u} + \alpha \cdot \nabla\mathbf{u}$$

with Dirichlet boundary conditions and $\epsilon \in \mathbb{R}_0^+$ and $\alpha \in \mathbf{R^2}$. Simple time-stepping methods are known to be unstable at large time-steps, and our exponential scheme suffers from similar problems, i.e. t cannot be taken too large. **Study the `MatLab` file `convection_diffusion.m`. Explain how the ODE is formed. What discretization is used? Add your `fAb` routine to the file. Compare your results with the original routine. Be thorough.**

3.2 The sign function

In control theory we are often interested in the eigenvalues λ of system matrices with $Re(\lambda) > 0$, since they correspond to unstable poles. In the design of controllers it is therefore interesting to have an efficient way to count the number of eigenvalues of a matrix in the right half-plane $Re(z) > 0$. Here we will build such a method.

Theorem 3 *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with k_- eigenvalues in the left plane, k_+ eigenvalues in the right plane and none on the imaginary axis, counting multiplicity. Let $sgn : \mathbb{C} \mapsto \{1, -1\}$ be defined by*

$$sgn(z) = \begin{cases} 1 & Re(z) \geq 0 \\ -1 & Re(z) < 0. \end{cases}$$

Then $trace(sgn(A)) = k_+ - k_-$.

Prove this. With this theorem in mind, interpret algorithm 1 (where we use `MatLab` notation for some of the operations). The function 'mode' here refers to the most frequently observed value. Be thorough.
Implement a computationally efficient version of algorithm 1. Apply your algorithm to the system `control.mat`. First plot the eigenvalues of the system using `eigs`. What do you observe? Do you expect Arnoldi to converge in few iterations? How would you remedy this? Hint: look at chapter 4 of [1] for inspiration. Compare the computed number of eigenvalues in the right half-plane to the actual number. Discuss the advantages and disadvantages of your scheme.

Algorithm 1: Computing $k_+(A)$

Data: $A \in \mathbb{C}^{n\times n}$, $d \in \mathbb{N}$, $N \in \mathbb{N}$
Result: $k_+ \in \mathbb{R}$
$\mathbf{q} \leftarrow \mathbf{0} \in \mathbb{R}^N$;
for $i = 1$ **to** N **do**
    $\mathbf{u} \leftarrow \mathrm{randn}(n, 1)$;
    $\hat{\mathbf{u}} \leftarrow \frac{\mathbf{u}}{\|\mathbf{u}\|}$;
    $[H, Q] \leftarrow \mathrm{arnoldi}(A, \hat{\mathbf{u}}, k)$;
    $q(i) \leftarrow \mathrm{trace}(\mathrm{sign}(H)))$;
end
$q := \mathrm{mode}(\mathbf{q})$;
$q \leftarrow (q + k)/2$

4 Quotation and questions

Points are awarded based on the correctness of your results and the quality of your code (40%), the insight demonstrated in your report (40%) and the organization and presentation of your report (20%). Don't hesitate to email me at simon.dirckx@kuleuven.be if anything is unclear or if you suspect something is wrong. If you are stuck on something you can always ask for a hint. This will be taken into account in the evaluation but will not severely impact your grade.

References

[1] Saad, Y.,*Numerical Methods for Large Eigenvalue Problems*, Society for Industrial and Applied Mathematics,2011