Scientific Computing 1st Assignment

Odysseas Lazaridis

September 2022

Every function can be found in the python file which I have uploaded. The only one which is not answered at all is the last subquestion of H. The name of the functions are the same as they is given in the instructions.

A Condition Number

A.1 Functions

The condition Matrix of a matrix M is given by the formula

$$\mathrm{cond}_\infty(M) = ||M||_\infty ||M^{-1}||_\infty, \tag{1}$$

In order to calculate the conditional number we need to calculate the max norm which is the formula

$$\mathrm{norm}_p(\mathbf{a}) = \left(\sum a_i^p\right)^{1/p}. \tag{2}$$

where for $p = \infty$ the upper equation changes to:

$$\mathrm{norm}_\infty(\mathbf{a}) = \max(\mathbf{a}). \tag{3}$$

By translating these formula to python, we have the following functions

```
def max_norm(M):
    sums = np.sum( np.abs(M),axis=1)
    return np.max(sums)

def condition_number(M):
    conditionNumber = max_norm(M)*max_norm(ReverseMatrix(M))
    return conditionNumber
```

A.2 Condition Number & Bound error

The error is given in the right hand side of our system with 8 significant digits. The bound of the relative error of x is given by the formula:

$$\frac{||\Delta\mathbf{x}||_\infty}{||\hat{\mathbf{x}}||_\infty} \leq \mathrm{cond}_\infty(\mathbf{E} - \omega\mathbf{S})\frac{||\Delta z||_\infty}{||z||_\infty}. \tag{4}$$

Where the the condition number will give us how many significant figures are gone:

$$\log_{10} ConditionNumber = MissingSignificantDigits \tag{5}$$

The condition numbers and the significant digits can be seen in the next table

ω	Condition Number	Significant Digits
0.8	327	5
1.146	152679	2
1.4	227	5

B Error in the Matrix

In this question, the difference with the previous one is that the error is given in the Matrix, which means that the bound for the relative error of x is given as:

$$\frac{||\Delta \mathbf{x}||_\infty}{||\hat{\mathbf{x}}||_\infty} \leq \mathrm{cond}_\infty(\mathbf{E} - \omega \mathbf{S}) \frac{||\delta\omega \mathbf{S}||_\infty}{||\mathbf{E} - \omega \mathbf{S}||_\infty}. \tag{6}$$

All the right hand side variables are known so we calculate the relative forward error and then the significant digits as shown bellow

```
sig_figs = -ceil(math.log10(rel_for_error))
```

The error bounds and the Significant digits can be seen in the next table

ω	Error Bounds	Significant Digits
0.8	0.00520774	2
1.146	2.40503	-1
1.4	0.00354736	2

C Matrix factorization and solving

In this exercise three separate functions were implemented which take an N matrix and they solve the system $Ax = b$.

C.1 `L,U = lu_factorize(M)`

The first function LU(A) takes as an input a matrix, A, and transforms it to an Upper triangular matrix U and an Lower triangular matrix L.

C.2 `y = forward_substitute(L, b)`

Forward substitution solves the problem $\mathbf{Ly} = \mathbf{b}$, with $\mathbf{L}$ a lower, triangular matrix and $\mathbf{y}$ unknown. The thought behind my algorithm is that you start solving the equations from top to bottom, by replacing all the results you found to the next equation.

C.3 `x = back_substitute(U,y)`

Backward-substitution is a method for obtaining the solution $\mathbf{x}$ in $Ux = y$, where $\mathbf{U}$ is a square upper triangular matrix. Here, I used the fact that back substitute is just a forward substitute with the matrix flipped upside down and left to right and also and the vector on the right side flipped upside down. So i made a function `problem_flipper(L,y)` which does what I just mentioned and feeds the output to forward substitute function

D Polarizability

D.1 `alpha = solve_alpha(omega)`

By applying the previous functions we get the following table

ω	$a(\omega)$	$a(\omega+\delta\omega)$	$a(\omega-\delta\omega)$	$\lvert a(\omega+\delta\omega)-a(\omega)\rvert$	$\lvert a(\omega)-a(\omega-\delta\omega)\rvert$
0.8	1.63614	1.64443	1.62782	0.008293	0.008322
1.146	2609.24	-4185.02	994.753	6794.253607	1614.482332
1.4	-2.70689	-2.69993	-2.71388	0.006966	0.006987

D.2 Correct Error Bound

The correct error-bound to understand the variation of the calculated polarizabilities due to the perturbation is (b). The reason behind that is that in (b) we examined the situation where the error is introduced inside the matrix and not on the right side of the system

D.3 Do your calculated values fall within this bound?

What we want, is to find an error bound so that we can evaluate the values of Δa_1 and Δa_2
The bound should be in the form of

$$||\Delta\alpha(\omega)||_\infty = B(\omega)|\delta\omega|. \tag{7}$$

To begin with we have that

$$||\Delta\alpha||_\infty = ||z^T\Delta x||_\infty \leq ||z||_\infty \, ||\Delta x||_\infty \tag{8}$$

and by multiplying (6) by $||x||_\infty$ and then substitute in (8) we end up with

$$||\Delta\alpha||_\infty \leq (||z||_\infty \, ||x||_\infty \, cond_\infty(E-)||S||_\infty \, /||E-)|\delta\omega||_\infty \tag{9}$$

Now we calculate the values and compare them with the theoretical upper bound

ω	$\lvert\lvert\Delta a\rvert\rvert_\infty$	Δa_1	Δa_2
0.8	0.014996	0.008293	0.00832
1.146	6187.330257	6794.253607	1614.482332
1.4	0.010420	0.006966	0.006987

It is easy to see that near the 1.146 is the only place that both of our Δ a are bigger than the upper bound that $||\Delta a||_\infty$ defines. As we will see at this point there is a singularity

E Plot $\alpha(\omega)$

For $\omega = 1.146307999$ we observe that the plot has a singularity which means that the Matrix is almost singular. It's easy to see that for very small changes of the ω(input) we have huge changes in the $a(\omega)$ (output).

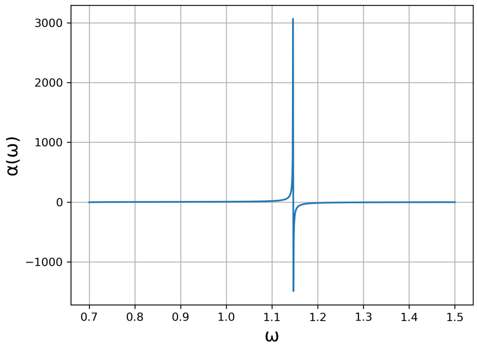

Figure 1: Polarizability for frequency range.

F Householder and least-squares

F.1 `Q,R = householder_QR_slow(A)`

`Q,R = householder_QR_slow(A)` takes a matrix M: $m \times n$ and uses the Householder method to compute its QR decomposition.

F.2 `V,R = householder_fast(A)`

Although I made the function and it works I didn't use it in the next steps. In the python file I print the combined VR matrix where R and v are combined in one matrix.

F.3 `x, r = least squares(A,b)`

`least_squares()` of an overdetermined system approximates the solution $\mathbf{x}$ to $\mathbf{Ax} = \mathbf{b}$. I implement this as

$$\mathbf{Ax} = \mathbf{b} \rightarrow \mathbf{QRx} = \mathbf{b} \rightarrow \mathbf{Q}^T\mathbf{QRx} = \mathbf{Q}^T\mathbf{b} \rightarrow \mathbf{Rx} = \mathbf{Q}^T\mathbf{b}, \mathbf{Q}^T\mathbf{b} = \mathbf{v} \rightarrow \mathbf{Rx} = \mathbf{v}$$

Apply backward substitution to obtain $\mathbf{x}$. The residual is the max-norm of the cropped part of $\mathbf{v}$.

G Approximating $\alpha(\omega)$ with simple polynomial

G.1 Choosing border

In this part we try to approximate our function with a polyonimic one. Sadly, $a(\omega)$ has a singularity around 1.146 so we can approximate our function only for omegas less than that number. I chose 1.13 for no particular reason. The closer this values goes to 1.146 the worse our approximations will be in the next questions.

G.2 4th grade polyonym

I split the area between [0.7,1.13] in 1000 parts with np.linspase and I applied my least squares routine for n = 4 and it resulted to the following coefficients:

a_0	a_1	a_2	a_3	a_4
229.8492	-1309.2133	2689.0557	-2362.6074	757.2896

G.3 Comparison between 4th and 6th grade polyonym

Here we repeat the same procedure but for n=6 and plot of the the magnitude of the relative error (in a log_{10}-scale) Next task is to calculate the mean value of each line to have a knowledge about the significant digits that each approximation has. As we can see from the relative error plot, for

significant figures for n=4	significant figures for n=6
0.6933	1.0765

$n = 6$ we have much better results. I didn't floor the results so we can see that indeed we have better results for n=6 as expected. These results are very sensitive to the choice of ω_p. When ω_p gets close to the singularity the approximation gets worse.From this diagram we also notice that are n peaks for nth-grade polynomym

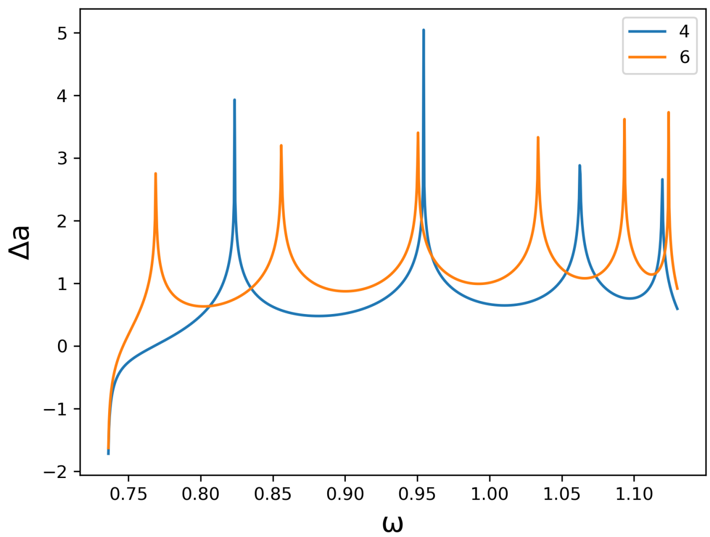

Figure 2: $-\log_{10}\text{rel.err}(P)$ for n=4 and n=6

H Approximating $\alpha(\omega)$ with rational polynomial

H.1 Failure of previous model

The approximation with a simple polyonymic function is guaranteed to fail because it can't describe the singularities of $a(\omega)$. The fractional polyonimic approximation, has a polyonym in the denominator which means that for its roots the values of P will be infinite and thus it is more suitable to describe such a model.

H.2 Fractional polyonym for $n = 2$ and $n = 4$

The problem that we will call our least squares algorithm to solve for the $n = 2$ approximation is the one bellow:
$$\begin{pmatrix} P_{00} & P_{01} & P_{02} & \dots & P_{0(2n+1)} \\ P_{10} & P_{11} & P_{12} & \dots & P_{1(2n+1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ P_{k0} & P_{k1} & P_{k2} & \dots & P_{k(2n+1)} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \\ -b_1 \\ -b_2 \\ \vdots \\ -b_n \end{pmatrix} = \begin{pmatrix} \mathrm{a}(\omega_0) \\ \mathrm{a}(\omega_1) \\ \vdots \\ \vdots \\ \vdots \\ \mathrm{a}(\omega_k) \end{pmatrix}$$

Where k is the number of our ω and P_{ij} is given from this weird formula I wrote only for the PDF.

$$P_{ij} = [\omega_i^{2-mod(j,2)} \alpha(\omega_i)^{0.5+0.5sign(n+1,5-j)}]^{0.5+0.5(sign(1.5-j)}$$

(10)

After applying the least squares formula for $n = 2$ I get the following coefficients We then apply

a_0	a_1	a_2	b_1	b_2
1.92786	-3.45141	1.14386	2.39353	-1.32701

again the least square formula for n=4 and we plot the relative error of both of them as can be seen in Figure 3. Here we observe a huge difference in the accuracy of each approximation, where for n=4 we seem to have 2 extra signifficant digits.

H.3 Extended approximation

By extending the range of the ω we can see more singularities apearing. This means that we should use a higher grade polyonym, where the grade of the polyonym should be at least as big as the number of the singularities. Sadly my cheap laptop couldn't plot in a reasonable amount of time the plots so I couldn't play as much as I wanted.

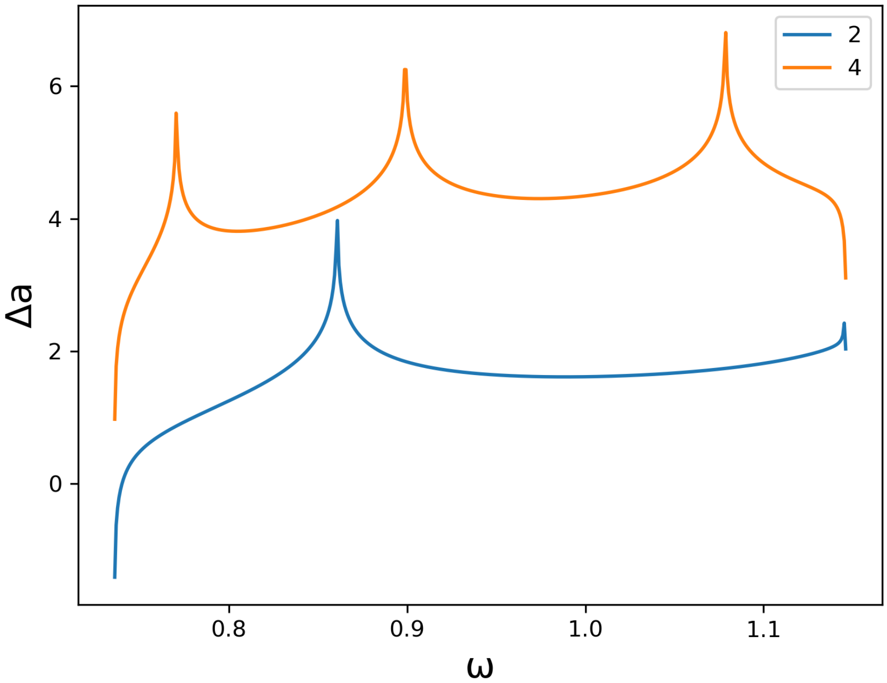

Figure 3: $-\log_{10}\mathrm{rel.err}(P) for n = 2 and n = 4$