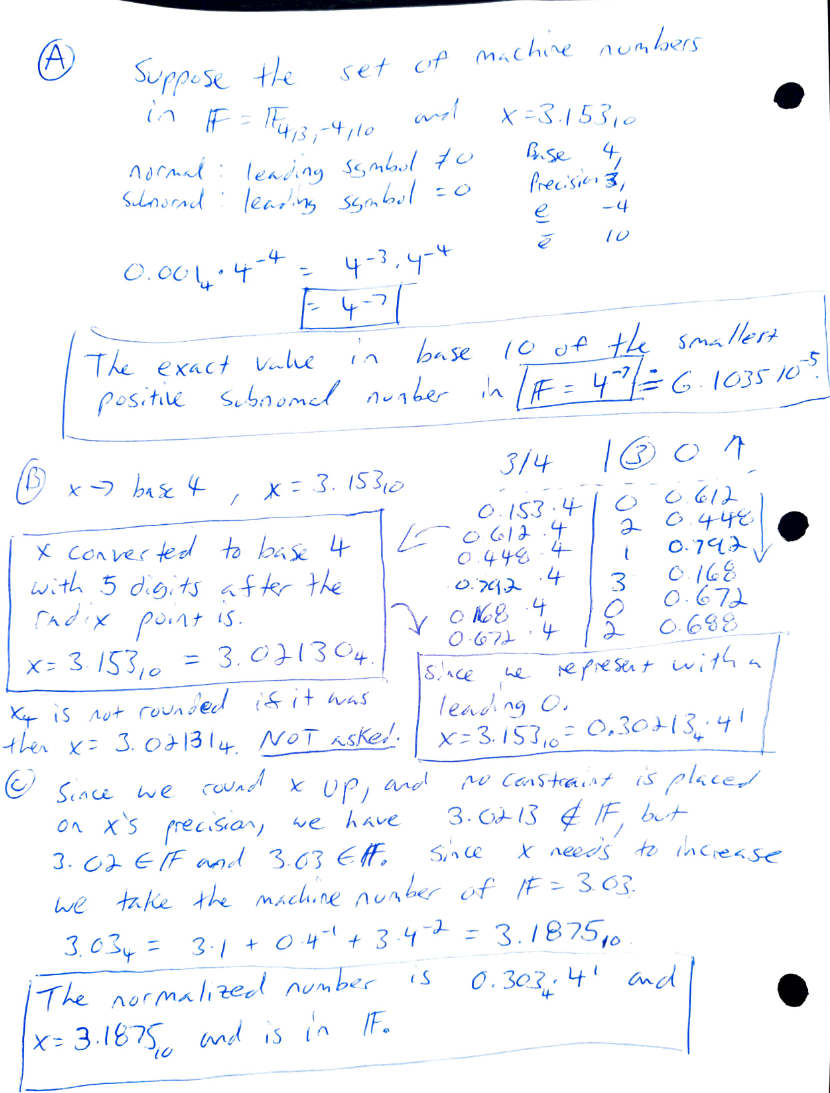
Math 3800 - A1

Nick Cooley

101021174

Date: 14 Jan 2024

Question 1.



No indication if this should be hand solved for use in MATLAB.

Question 2.

Initially I understood this question to solve for the value x in A = [-1 4; 3 x] such that the l2 norm of A is equal to x. This doesn’t exist. The algebra showing it is in the A1of101021174.m.

Ms Nookala and Professor Cheung both clarified that the question is asking us to find a value where norm(A(x), 2) ~= sqrt(max(eig(ATA(x))).

I have trust issues with number types and type casting into Booleans at the wrong time. I selected a very small threshold close enough to zero as a tolerance.

We start with two empty arrays. The script loops from 1 to 100 in the naturals. Append on top of arr0 (where the difference is sufficiently zero) and arrn0 (difference not sufficiently zero). While looping we print where the where the difference is nonzero.

After the loop is complete, we loop over both arr0 and arrn0 indicating the values x and the difference between the two L2 norm calculation methods.

I included all solutions and not the first one as it’s just better practice.

I wound an almost 50-50 even spread of the difference between our norm(A, 2) and sqrt(max(eig(ATA(x)))) where it exists is miniscule over all x in [1, 100]. This begs to ask the question which calculation is more correct?

Question 3.

We compare the differences in y and y\_ (estimate) with our two termed Taylor series to find the forward largest error.

A screen shot of a graph

Description automatically generated

Figure 1. Plot of real(red) and estimated (blue) values of cos on the interval [0, 0.5235] with 244 evenly spaced plots.

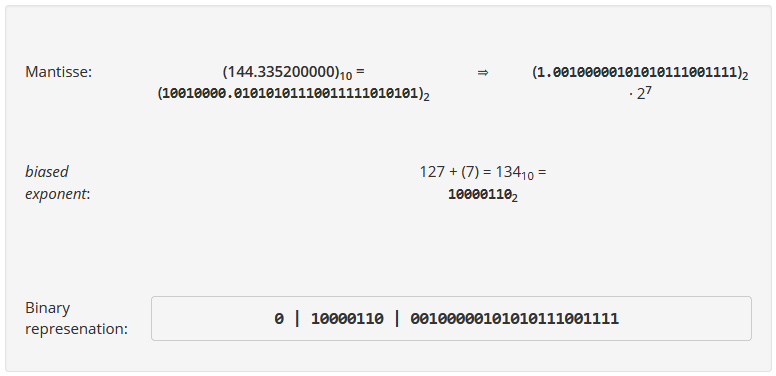
The maximum difference between the real cosine vector and Taylor series cosine vector to 2 terms is: 0.003100912358768. This occurs at the most extreme value x due to the error in quadratic form missing infinite many terms that approximate cosine. The infinite Taylor series representation has x in the numerator raised to increasing powers. x = 0 cancels them all out hence no error at x = 0.

The forward error is: -0.00015056 relative forward error is: -0.00015521 taken as the difference between y and estimated y at point x = 0.2453.

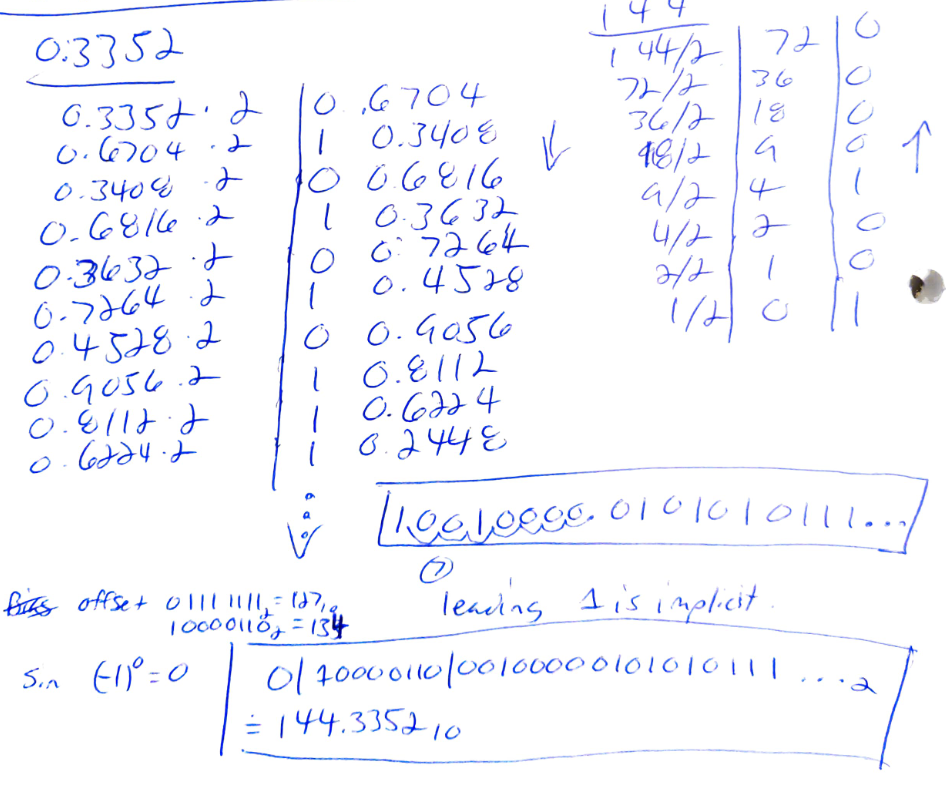
The backwards error is: 0.00061921 and relative backwards error is: 0.00252430. The backwards error is the difference between given x and estimated x\_ is calculated by taking the arccos(y\_).

Question 4.

See next page.



We were not asked to solve this, but I wanted to practice.



Acknowledgements:

1. Verifying correct conversions.

<https://meteorconverter.com/conversions/number-bases>

1. Converting decimals to other bases with radix.

<https://www.youtube.com/watch?v=96MJVzVKoIE>

1.c Trying to understand how to handle normalized leading in ambiguous units.

<https://en.wikipedia.org/wiki/Normalized_number>

2. Syntax on output streams.

<https://www.mathworks.com/help/matlab/matlab_prog/formatting-strings.html>

4. Assigned resource.

<https://www.uibk.ac.at/mechatronik/mikroelektronik/lehre/webapps/ieee754.html.en>