Assignment - 1

1. Condition Number

derivation of formula

let $f: R \to R$ [function from R to R]

$x \to f(x)$

Let $\hat{x} = x + \Delta x$ where Δx is a small pertubation in x

$f(\hat{x}) = f(x+\Delta x)$

$$\text{Relative conditional number} = \frac{\left|\frac{f(x+\Delta x) - f(x)}{f(x)}\right|}{\left|\frac{(x+\Delta x) - x}{x}\right|}$$

$$= \left|\frac{x}{\Delta x}\right| \frac{1}{|f(x)|} |f(x+\Delta x) - f(x)| \quad \text{--- (1)}$$

using talyor expansion for $f(x+\Delta x)$ we get

$$f(x+\Delta x) - f(x) \approx \Delta x f'(x)$$

replacing value in (1)

$$\left|\frac{x}{\Delta x}\right| \frac{1}{|f(x)|} |\Delta x f'(x)|$$

$$= \left|\frac{x f'(x)}{f(x)}\right|$$

$$K(x) = \left|\frac{x f'(x)}{f(x)}\right|$$

(a) $y - a^x = 0$

$y = a^x = f(x)$

$\frac{dy}{dx} = f'(x) = \frac{d}{dx} a^x$

let $a^x = t$

$\log t = x \log a$

$\frac{d}{dx} \log t = \frac{d}{dx} x \log a$

$\frac{1}{t} \frac{dt}{dx} = \log a$

$\frac{dt}{dx} = t \log a$

$\frac{dt}{dx} = a^x \log a$

$\frac{dy}{dx} = f'(x) = a^x \log a$

hence by the formula

$K(x) = \left| \frac{x f'(x)}{f(x)} \right|$

$= \left| \frac{x \cdot a^x \log a}{a^x} \right|$

$\boxed{K(x) = |x \log a|}$

b. $x + 1 - y = 0$

$y = x + 1$

$\frac{dy}{dx} = f'(x) = 1$

By the formula

$K(x) = \left| \frac{x f'(x)}{f(x)} \right|$

$\boxed{K(x) = \left| \frac{x}{x+1} \right|}$ or $\boxed{K(x) = \left| \frac{1}{1 + 1/x} \right|}$

2. Given: Vector norms $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$

To prove : $\big| \|x\| - \|y\| \big| \leq \|x-y\|$

Proof: For us to show $\big| \|x\| - \|y\| \big| \leq \|x-y\|$ it is suffecient to show these 2 condition

a. $\|x-y\| \geq \|x\| - \|y\| \quad \forall\ x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$

b. $\|x-y\| \geq \|y\| - \|x\| \quad \forall\ x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$

Case-I To prove: $\|x-y\| \geq \|x\| - \|y\|$

$x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ [given]

$\Rightarrow x - y \in \mathbb{R}^n$

$\|x-y\| + \|y\| \geq \|x - \cancel{y} + \cancel{y}\|$ [Triangular inequality Of vector norms]

$\Rightarrow \|x-y\| + \|y\| \geq \|x\|$

$\Rightarrow \|x-y\| \geq \|x\| - \|y\| \quad — (1)$

Case-II To prove : $\|x-y\| \geq \|y\| - \|x\|$

$x \in \mathbb{R}^n$ & $y \in \mathbb{R}^n$ [given]

$\Rightarrow y - x \in \mathbb{R}^n$

Hence

$\|y-x\| + \|x\| \geq \|y - x + x\|$ [Triangular inequa

$\Rightarrow \|y-x\| + \|x\| \geq \|y\|$

$\Rightarrow \|y-x\| \geq \|y\| - \|x\|$

$\Rightarrow \|x-y\| \geq \|y\| - \|x\|$ [Homogeneity principle]

$— (11)$

$$\left[\begin{aligned} \|y-x\| &= \|(-1)(x-y)\| \\ &= |-1|\,\|x-y\| \\ &= \|x-y\| \end{aligned}\right]$$

Hence from (1) & (11) we can say

$\big| \|x\| - \|y\| \big| \leq \|x-y\|$

$\Rightarrow$ Vector norms are Lipschitz continuous

Hence proved

Ans-3

(a) Given: $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ & $x \otimes y = \begin{bmatrix} x y_1 \\ \vdots \\ x y_m \end{bmatrix}_{mn}$

To find: value of $\|x \otimes y\|$ in terms of $\|x\|_p$ & $\|y\|_p$ for $p = 1, 2$ and ∞

for $p = 1$

for $x \in \mathbb{R}^n$ $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$

$$\|x \otimes y\|_p = \{ |x_1 y_1| + |x_1 y_2| + |x_1 y_3| \cdots + |x_1 y_m| + $$
$$|x_2 y_1| + |x_2 y_2| + |x_2 y_3| \cdots + |x_2 y_m| +$$
$$\vdots \quad \vdots \quad \vdots \quad \vdots$$
$$|x_n y_1| + |x_n y_2| + |x_n y_3| \cdots + |x_n y_m|$$

$\Rightarrow |y_1| (|x_1| + |x_2| + \cdots |x_n|) + |y_2| (|x_1| + |x_2| + \cdots |x_n|) +$
$\cdots + |y_n| (|x_1| + |x_2| + \cdots |x_n|)$

$\Rightarrow (|x_1| + |x_2| + \cdots |x_n|)(|y_1| + |y_2| + \cdots + |y_n|)$

$\Rightarrow \left(\sum_{i=1}^{n} |x_i| \right) \left(\sum_{j=1}^{m} |y_j| \right)$

$\Rightarrow \|x\| \|y\|$ [from defination]

$$\boxed{\|x \otimes y\| = \|x\| \|y\|}$$

for $p=2$

$x \in \mathbb{R}^n \; ||x||_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$ and $y \in \mathbb{R}^m \; ||y||_2 = \left(\sum_{j=1}^{m} |y_j|^2\right)^{1/2}$

similarly for

$$||x \otimes y||_2 = \{ |x_1 y_1|^2 + |x_1 y_2|^2 + \cdots + |x_1 y_m|^2 + |x_2 y_1|^2 + |x_2 y_2|^2 + \cdots + |x_2 y_m|^2 + \cdots + |x_n y_1|^2 + |x_n y_2|^2 + \cdots + |x_n y_m|^2 \}^{1/2}$$

$$\Rightarrow \{ |y_1|^2 (|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2) + |y_2|^2 (|x_1|^2 + |x_2|^2 + \cdots |x_n|^2) + \cdots + |y_m|^2 (|x_1|^2 + |x_2|^2 + \cdots |x_n|^2) \}^{1/2}$$

$[\, |x_i y_j|^2 = |x_i|^2 |y_j|^2 \,]$

$$\Rightarrow \{ (|x_1|^2 + |x_2|^2 + \cdots |x_n|^2)(|y_1|^2 + |y_2|^2 + \cdots |y_m|^2) \}^{1/2}$$

$$\Rightarrow (|x_1|^2 + |x_2|^2 + \cdots |x_n|^2)^{1/2} (|y_1|^2 + |y_2|^2 + \cdots |y_m|^2)^{1/2}$$

$$\Rightarrow \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \left(\sum_{j=1}^{m} y_j^2\right)^{1/2}$$

$$\Rightarrow ||x||_2 \, ||y||_2$$

$$\boxed{||x \otimes y||_2 = ||x||_2 \, ||y||_2}$$

for $p = \infty$

$\|x \otimes y\|_\infty = \max(\{|x_i y_j|$ wher $1 \le i \le n$ & $1 \le j \le m)$

$\Rightarrow \max(|x_i||y_j|) \; \forall \; i \in 1$ to n & $\forall j \in 1$ to m

$\Rightarrow (\max(|x_i|) \; \forall \; i \in 1$ to $n)\,(\max |y_j| \; \forall j \in 1$ to m

$\Rightarrow \|x\|_\infty \|y\|_\infty$ $[\because \|x\|_\infty = \max(|x_i|)$ $\forall i \; 1 \le i \le n$

$$\boxed{\|x \otimes y\|_\infty = \|x\|_\infty \|y\|_\infty}$$

DATE
PAGE

(b)

for $p = 1$
for $A \in R^{m \times n}$ $\quad ||A||_1 = \max_j \left(\sum_{i=1}^{m} |a_{ij}| \right)$ [iterating over j and find max]

for $j=1$ ie first column the value of 1 norm is

$$\{ |A_{11} B_{11}| + |A_{21} B_{11}| + \cdots |A_{m1} B_{11}| +$$
$$|A_{11} B_{21}| + |A_{21} B_{21}| + \cdots |A_{m1} B_{21}| +$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$|A_{m1} B_{k1}| + |A_{21} B_{k1}| + \cdots + |A_{m1} B_{k1}| \}$$

this can be simplified as:

$$\Rightarrow (|A_{11}| + |A_{21}| + \cdots |A_{m1}|)(|B_{11}| + |B_{21}| + \cdots + |B_{k1}|)$$
$$\Rightarrow \sum_{i=1}^{m} |A_{i1}| \sum_{j=1}^{k} |B_{j1}|$$

similarly for $j=2$ ie 2^{nd} colum $\rightarrow \sum_{i=1}^{m} |A_{i2}| \sum_{j=1}^{k} |B_{j2}|$
" " $j=3$ ie 3^{r} colum $\Longrightarrow \sum_{i=1}^{m} |A_{i3}| \sum_{j=1}^{k} |B_{j3}|$

$\vdots \qquad \vdots$

In general 1 norm of any col = (1 norm of col of A) $\otimes$ (1 norm of col of B)

$\Rightarrow$ max 1 norm of cols of $A \otimes B$ = max[(1 norm of col of A) $\otimes$ (1 norm of col of B)]

$\Rightarrow$ max 1 norm of cols of $A \otimes B$ = max(1 norm of col of A) $\otimes$ max(1 norm of col of B)

$$\boxed{||A \otimes B|| = ||A||_1 \, ||B||_1}$$

where $A \in R^{m \times n}$ & $B \in R^{k \times l}$

for $p = \infty$
similar to previous one
∞ norm of any column = $\left(\begin{array}{c}\infty \text{ norm of } A \\ \text{in that col}\end{array}\right) \otimes \left(\begin{array}{c}\infty \text{ norm of} \\ B \text{ in that col}\end{array}\right)$

$$\Rightarrow \|\text{row } j \text{ of } A\|_\infty \otimes \|\text{row } j \text{ of } B\|_\infty$$

$$\max_{i} \|\text{row } i \text{ of } A \otimes B\| = \max \left[\|\text{row } j \text{ of } A\|_\infty \otimes \|\text{row } j \text{ of } B\|_\infty \right.$$

$$\Rightarrow \max (\|\text{row } j \text{ of } A\|) \otimes \max \|\text{row } j \text{ of } B\|_\infty$$

$$\Rightarrow \|A\|_\infty \otimes \|B\|_\infty$$

$$\boxed{\|A \otimes B\| = \|A\|_\infty \otimes \|B\|_\infty}$$ where $A \in \mathbb{R}^{m \times n}$ & $B \in \mathbb{R}^{k \times t}$

DATE
PAGE

4. Given: $A \in R^{n \times n}$ is a nilpotent matrix with index = 2
To prove: A is singular

Proving Technique: Contradiction

Proof: Let A have linear independent colums

$\because$ A has linearly independent colums
$\Rightarrow Ax = 0$ and A is invertible — (I)
also $A^2 = 0$ [given to us as A is nilpotent of index 2
$\Rightarrow A^2 = 0$] — (II)
from statement (I) & (II) we get
$Ax = A^2$ — (IV)
$\because$ A is linearly independent ~~colums~~
$\Rightarrow$ A is invertible — (V)
~~fro~~
$Ax = A^2$ [from (IV)]
pre multiplying with A^{-1} [$\because$ A is invertible from statement (V)]

$A^{-1}Ax = A^{-1}A^2$
$(A^{-1}A)x = A$
$Ix = A$ where I is an identity matrix
$x = A$ — (VI)
But $x = 0$ since $Ax = 0$ only have trivial sol^n.
This is a contradiction [$\Rightarrow \Leftarrow$]
$\Rightarrow$ A is a linearly dependent matrix
$\Rightarrow$ A is not invertible [Invertibility theorem]
$\Rightarrow$ A is a singular matrix
Hence proved.

5. Given: Matrices A and B s.t $A, B \in \mathbb{R}^{n \times n}$ and A & B are nonsingular

To prove: $k(AB) \leq k(A)\,k(B)$

Proof: for a matrix A which is nonsingular & $\in R^{n \times n}$

$k(A) = ||A||\,||A^{-1}||$

similarly $k(B) = ||B||\,||B^{-1}||$

$k(A)\,k(B) = ||A||\,||A^{-1}||\,||B||\,||B^{-1}||$ —— (I)

also $k(AB) = ||AB||\,||(AB)^{-1}||$

$\Rightarrow ||AB||\,||B^{-1}A^{-1}||$ —— (IV)

also $||AB|| \leq ||A||\,||B||$ —— (II) [submultiplicative property]

similarly $||B^{-1}A^{-1}|| \leq ||B^{-1}||\,||A^{-1}||$ —— (III)

from (IV) (II) & (III) we get

$||AB||\,||B^{-1}A^{-1}|| \leq ||A||\,||B||\,||A^{-1}||\,||B^{-1}||$

$\Rightarrow k(AB) \leq ||A||\,||B||\,||A^{-1}||\,||B^{-1}||$ [from (IV)]

$\Rightarrow k(AB) \leq ||A||\,||A^{-1}||\,||B||\,||B^{-1}||$

$\Rightarrow k(AB) \leq k(A)\,k(B)$ [from (I)]

$k(AB) \leq k(A)\,k(B)$

Hence proved

Answer-6

a. For the first part the output is as follows:

```
0.5
0.25
0.125
0.0625
0.03125
0.015625
0.0078125
0.00390625
0.001953125
0.0009765625
0.00048828125
0.000244140625
0.0001220703125
6.103515625e-05
3.0517578125e-05
1.52587890625e-05
7.62939453125e-06
3.814697265625e-06
1.9073486328125e-06
9.5367431640625e-07
4.76837158203125e-07
2.384185791015625e-07
1.1920928955078125e-07
5.960464477539063e-08
2.9802322387695312e-08
1.4901161193847656e-08
7.450580596923828e-09
```

```
2.590327e-318
1.295163e-318
6.4758e-319
3.2379e-319
1.61895e-319
8.095e-320
4.0474e-320
2.0237e-320
1.012e-320
5.06e-321
2.53e-321
1.265e-321
6.3e-322
3.16e-322
1.6e-322
8e-323
4e-323
2e-323
1e-323
5e-324
0.0
```

We are dividing 1 by 2 in each iteration of the loop and we are looping till the input doesn't become zero. The 64-bit machine follows Double Precision IEEE 756 floating-point standards in which the numbers are stored in 64 bits. Since we can continuously divide 1 infinite times till we reach 0 but in double precision, we can divide up to a minimum number, and any number smaller than that number would be considered as 0. This is called the underflow of the floating-point number.

b. For the second part the input is as follows:

```
1.1102230246251565e-16
```

In the code above the value of a is not changing and is always 1.0. Hence the code can be simplified as follows

```
[ ]  eps = 1.
     b = 1. + eps
     while b!=1.0:
       eps /=2
       b = 1. + eps
     print(eps)

     1.1102230246251565e-16
```

In the above simplified code, we are dividing eps by 2 continuously till the value of 1.0+eps does not become equal to 1.0. This is only possible when the value of eps becomes 0. The machine epsilon value for Double-precision is 2.220446049250313e-16. The value of eps at the end of the loop is 1.1102230246251565e-16. If further computation is carried out(let say hypothetically) then the value of eps would get in the order of 10^-17 which is smaller than machine epsilon and can't be represented by the machine and hence would be rounded to 0.

c. The output of the third part is as follows:

```
2.0
4.0
8.0
16.0
32.0
64.0
128.0
256.0
512.0
1024.0
2048.0
4096.0
8192.0
16384.0
32768.0
```

```
2.1944496275174755e+304
4.388899255034951e+304
8.777798510069902e+304
1.7555597020139804e+305
3.511119404027961e+305
7.022238808055922e+305
1.4044477616111843e+306
2.8088955232223686e+306
5.617791046444737e+306
1.1235582092889474e+307
2.247116418577895e+307
4.49423283715579e+307
8.98846567431158e+307
inf
```

The explanation is similar to the first part. In Double Precision IEEE 756 floating point standards the numbers are stored in 64 bits and the largest possible number which can be stored by a machine is 1.7976931348623157e+308. Any number greater than this would be considered as infinity. Hence in the code where we are doubling the number by 2 the largest possible number will be 8.98846567431158e+307 and after this the numbers would overflow.

Answer-7

a. The most accurate approximation of e is obtained when the value of n = 10^8. The value of relative error is 1.1077470720850393e-08. Ideally, the approximation value should increasingly become more accurate on increasing the value of n. But in this case, the max accuracy is achieved for n = 10^8 which is not the max value of n considered(max value considered was n = 10^{15}). Let us try to understand with the help of an example:

   Consider n = 10^{15}

   Using the formula y = $(1 + (1/n))^n$ for n = 10^{15}. On computing with computer we get y = 3.035035206549262 [Computed Using Python3] --------- (1)

   Machine Epsilon(ε_M) is a fixed number such that for any x $\in$ R and its fl(x) $\in$ F we have $|x - fl(x)|/|x|$ = O(ε_M) ---------- (2)

   For 64bit computer the the value of ε_M = 2.220446049250313e-16 [Double point precision computed using python] ----------- (3)

   Substituting the value of (1) and (3) in equation (2) and using equality we get the value of x = 2.7182818284590446 ≈ e

   Hence the precision is hampered due to machine epsilon. For the value n = 10^{15} the calculated value comes out to be $\left(1 + \frac{1}{10^{15}}\right)^{10^{15}}$ = 2.7182818284590 (calculated using wolfram alpha) the value is rounded off to 3.035035206549262 because of machine epsilon. Machine epsilon can be understood as the unit roundoff.

b. The stopping criteria can be understood as follows: we are calculating 1/(x!) where 1 <= x < ∞. But in Double point precision the machine epsilon(ε_M) value is 2.220446049250313e-16. Hence any value smaller than ε_M would be considered as zero. So on reaching the smallest possible value that can be stored by machine the further values would become 0 and the loop will break