Assignment 1

Purvam Jain EE20B101

November 13, 2023

Contents

1 Question 1

1.1 How many non-zero Floating Point numbers (FPN) can be represented by our machine?

We can try solving this by considering the number of choices we have for the values S,A,B,C and E which will result in a non-zero floating point number.
For a non-zero floating point number,

- **S** can take one of a total of 2 permissible values, namely $\{0, 1\}$. S taking any other value is used to represent NaN which is not a floating point number.
- **A** can take one of a total of 9 permissible values, namely$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. A is not permitted to take the value 0, as that is a requirement to represent zero on the machine.
- **B**,**C** and **E** can take one of a total of 10 permissible values, namely $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ as there is no restriction on them when it comes to representing zero on the machine thereby permitting all possible choices.

Employing the *fundamental principle of counting*,the total number of ways in which we can represent a non-zero floating point number is the product of the number of ways in which each individual value S,A,B,C,E can be represented for the given purpose.
Therefore, the number of non-zero Floating Point Numbers that can be represented on the machine is 2x9x10x10x10 = $18\text{x}10^3$ or 18000 numbers of which 17998 are normal floating point numbers and the other two are $+\infty$ and $-\infty$.

1.2 How many FPN are in the following intervals?

1.2.1 (9, 10)

We can try employing the same approach as in the previous subdivision to find the total number of FPN.

- **Representation of the lower bound 9**:
  To find the machine representation of 9, let us express it in scientific notation with 3 significant digits (as that is the extent to which the machine can represent the number).
  $9 = 9.00\text{x}10^0$
  From this we can see that 9 is represented in the below form.

0	9	0	0	5

  - S is 0 as 9 is positive.

- A is 9 as it is the first digit(digit before the decimal point) in scientific notation.
- B and C are 0 as they're the first two significant digits that trail the decimal point.
- E is 5 as the exponent in scientific notation is 0. Adding the bias 5 to the exponent value, we get the value of E to be 5.

- **Representation of the upper bound 10**:
  Expressing 10 in scientific notation to three significant digits, we get,
  $10 = 1.00\text{x}10^1$
  From this we can see that 10 is represented in the below form.

0	1	0	0	6

  - S is 0 as 10 is positive.
  - A is 1 as it is the first digit(digit before the decimal point) in scientific notation.
  - B and C are 0 as they're the first two significant digits that trail the decimal point.
  - E is 6 as the exponent in scientific notation is 1. Adding the bias 5 to the exponent value, we get the value of E to be 6.

- **Identifying restrictions on S,A,B,C,E through the above two representations**

  - As both 9 and 10 are positive, any number between them is also positive therefore, S remains 0. Therefore, the possible values of S includes only 0.
  - As the interval is an open one, meaning 9 and 10 are not included, the term A necessarily has to take the value 9 as anything else would not represent floating point numbers lying in the interval as the largest FPN below 10 starts with the digit 9 and the smallest FPN above 9 also starts with the digit 9.Therefore, the possible values of A includes only 9.
  - B and C can take any value with the sole exception of both of them being 0, ie, $\{B, C\} = \{0, 0\}$ as that would represent 9 which doesn't lie inside the interval.Therefore, the possible values of B,C include those from $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ obeying the constraint $\{B, C\} \neq \{0, 0\}$.
  - E necessarily has to take the value 5 as the E term remains constant throughout the interval and only changes at the endpoint 10 which is not part of the interval.Therefore, the possible values of E includes only 5.

- **Computing the answer**
  Therefore by the *fundamental principle of counting*, the total number of floating point numbers lying between 9 and 10 are, 1x1x(10x10-1)x1 = 99 FPN. (10x10 - 1 represents the total number of values the pair $\{B, C\}$ can take without violating $\{B, C\} \neq \{0, 0\}$ as B and C can each take a total of 10 possible values individually)

1.2.2 (10, 11)

- **Representation of the lower bound 10**:
  Expressing 10 in scientific notation to three significant digits, we get,
  $10 = 1.00\text{x}10^1$
  From this we can see that 10 is represented in the below form.

0	1	0	0	6

  - S is 0 as 10 is positive.
  - A is 1 as it is the first digit(digit before the decimal point) in scientific notation.
  - B and C are 0 as they're the first two significant digits that trail the decimal point.
  - E is 6 as the exponent in scientific notation is 1. Adding the bias 5 to the exponent value, we get the value of E to be 6.

- **Representation of the upper bound 11**:
  Expressing 10 in scientific notation to three significant digits, we get,
  $11 = 1.10\text{x}10^1$
  From this we can see that 11 is represented in the below form.

0	1	1	0	6

  - S is 0 as 11 is positive.
  - A is 1 as it is the first digit(digit before the decimal point) in scientific notation.
  - B is 1 and C is 0 as they're the first two significant digits that trail the decimal point.
  - E is 6 as the exponent in scientific notation is 1. Adding the bias 5 to the exponent value, we get the value of E to be 6.

- **Identifying restrictions on S,A,B,C,E through the above two representations**

- As both 10 and 11 are positive, any number between them is also positive therefore, S remains 0. Therefore, the possible values of S includes only 0.
- As the interval is an open one,the term A necessarily has to take the value 1 as anything else would not represent floating point numbers lying in the interval as the largest FPN below 11 starts with the digit 1 and the smallest FPN above 10 also starts with the digit 1.Therefore, the possible values of A includes only 1.
- B necessarily has to take the value 0 due to the same reason above for A applied to the second significant digit. C however can take any value besides 0 as that would represent 10 which isn't part of the interval. Therefore, the possible values of B includes only 0 and that of C is $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- E necessarily has to take the value 6 as the E term remains constant throughout the interval.Therefore, the possible values of E includes only 5.

- **Computing the answer**
  Therefore by the *fundamental principle of counting*, the total number of floating point numbers lying between 10 and 11 are, 1x1x1x9x1 = 9 FPN.

1.2.3 (0, 1)

- **Representation of the lower bound 0**:
  From the table, we can see that +0 (as the interval is between 0 and 1) is represented in the below form.

0	0	X	X	X

  Where X indicates that any digit can be put in that place.

- **Representation of the upper bound 1**:
  Expressing 1 in scientific notation to three significant digits, we get,
  $1 = 1.00\text{x}10^0$
  From this we can see that 1 is represented in the below form.

0	1	0	0	5

  - S is 0 as 1 is positive.
  - A is 1 as it is the first digit(digit before the decimal point) in scientific notation.
  - B and C are 0 as they're the first two significant digits that trail the decimal point.

- E is 5 as the exponent in scientific notation is 0. Adding the bias 5 to the exponent value, we get the value of E to be 5.

- **Identifying restrictions on S,A,B,C,E through the above two representations**

  - As we're looking at the interval between 0 and 1 which comprises only positive numbers, S remains 0. Therefore, the possible values of S includes only 0.
  - As the interval is an open one,the term A cannot take the value 0 as 0 is an endpoint of the interval. A can take any other value since with an appropriate exponent term, the number will remain in the interval. Therefore, the possible values of A are $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
  - B and C can take any value whatsoever as there is no inherent constraint on them here as in the previous parts. For any value of B and C, with an appropriate A and E value, the number will remain in the interval.Therefore, the possible values of B,C are $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
  - E has to take values less than 5. If E takes the value 5, it automatically represents a number lying outside the interval. This is because A is the digit appearing before the decimal point in the scientific notation, and for any permitted value of A, the FPN will be greater than or equal to 1 for an E value of 5 or higher. Additionally, the values of E less than 5 belong to numbers with negative exponents which lie in the given interval. Therefore, the possible values of E are $\{0, 1, 2, 3, 4\}$.

- **Computing the answer**
  Therefore by the *fundamental principle of counting*, the total number of floating point numbers lying between 0 and 1 are, 1x9x10x10x5 = 4500 FPN.

1.3 Identify the smallest positive and largest positive FPN on the machine.

Smallest positive FPN:
For the smallest positive FPN, we would like the following:

- The number should be one with the largest negative exponent in order to be as small as possible.Therefore, we have E = 0, thereby the actual exponent being -5 (subtracting the bias).

- A,B,C have to take the values 1,0,0 respectively as for a given exponent E, $1.00\text{x}10^E$ is smaller than any other number with the same exponent.

- S has to take the value 0 as the number is positive.

With these, the representation of the smallest positive FPN on the machine is

0	1	0	0	0

with its value being 1.00x10^{-5}.

Largest positive FPN:

If the question is referring to the largest positive FPN, then it is $+\infty$.

However in the case it is referring to the largest positive *normal* FPN, meaning finite floating point number we should tread as we did for the smallest positive FPN.

For the largest positive normal FPN, we would like the following:

- The number should be one with the largest positive exponent in order to be as large as possible.Therefore, we have E = 9, thereby the actual exponent being +4 (subtracting the bias).
- A,B,C have to take the values 9,9,8 respectively as for a given exponent E, 9.98x10^{E} is larger than any other number with the same exponent(considering only 3 significant digits) with the sole exception being 9.99x10^{E}.However,the exception requires A,B,C,E = 9 which happens to be the representation of ∞ in our machine and hence this combination is not permitted to represent a normal FPN.
- S has to take the value 0 as the number is positive.

With these, the representation of the largest positive FPN on the machine is

0	9	9	8	9

with its value being 9.98x10^{4}.

1.4 Identify the machine precision.

Machine precision ϵ_m is defined as the difference between the number 1 and the smallest number exceeding 1 represented on the machine.

Referring to a previous subdivision, we can see that 1 is represented as

0	1	0	0	5

in the machine.

We are now required to find the next number greater than 1 which is representable on the machine.For this, we should keep the values of S,A,B and E intact and increase C by the smallest possible step. This is because, changing S,A or E will make the number drastically larger or smaller than 1 which defeats our purpose. Since C represents the number appearing after B in the decimal point,it is more sensitive to change when compared to B.

Therefore, the number just greater than 1 representable on the machine is

0	1	0	1	5

with a value of 1.01x10^{0}.

Therefore, employing the definition of machine precision, ϵ_m = 1.01 - 1 = 10^{-2}

1.5 What is the smallest positive integer not representable exactly on this machine?

Using the fact that any number with more than three significant digits gets stripped down to its first three significant digits in our machine,logically, the smallest positive integer not representable exactly would be the smallest one with more than 3 significant digits.
This number would be 1001 as in scientific notation, it is expressed as 1.001x10^3, but in our machine this gets chopped off to 1.00x10^3 and hence is not representable exactly.
Any positive integer below 1001 can be expressed accurately using just three significant digits and is therefore representable exactly on the machine.

1.6 Consider solving the following recurrence on our machine:

$a_{n+1} = 5a_n - 4a_{n-1}$

- **Initialization:**
  $a_1, a_2 = 2.932$. As this has 4 significant digits, in our machine, a_1, a_2 will be chopped down to 3 significant digits and represented as | 0 | 2 | 9 | 3 | 5 | with a value of 2.93x10^0.

- **n = 2:**
  - **Multiplication**
    As n = 2, a_n is a_2 represented as 2.93x10^0 on the machine. Therefore the representation of 5xa_n will be 5x2.93x10^0 = 1.465x10^1 chopped to three significant digits which is 1.46x10^1.
    Similarly doing the same for the a_{n-1} term a_1, we have, 4xa_1 4x2.93x10^0 = 1.172x10^1 represented as 1.17x10^1 on the machine.
  - **Subtraction**
    Subtracting the machine representation of $4a_1$ from that of $5a_2$ we get 2.90x10^0 which can be accurately represented in the machine as it has three significant digits.
  - **Update**
    Therefore, a_3 will be represented as | 0 | 2 | 9 | 0 | 5 | with a value of 2.90x10^0.

- **n = 3:**
  - **Multiplication**
    As n = 3, a_n is a_3.Therefore the representation of 5xa_n will be 5x2.90x10^0 = 1.45x10^1 which already has 3 significant digits and therefore no chopping takes place.

Similarly doing the same for the a_{n-1} term a_2, we have, 4xa_2 4x2.93x10^0 = 1.172x10^1 represented as 1.17x10^1 on the machine.

- **Subtraction**
  Subtracting the machine representation of $4a_2$ from that of $5a_3$ we get 2.80x10^0 which can be accurately represented in the machine as it has three significant digits.
- **Update**
  Therefore, a_4 will be represented as | 0 | 2 | 8 | 0 | 5 | with a value of 2.80x10^0.

- **n = 4:**
  - **Multiplication**
    As n = 4, a_n is a_4. Therefore the representation of 5xa_n will be 5x2.80x10^0 = 1.40x10^1 which already has 3 significant digits and therefore no chopping takes place.
    Similarly doing the same for the a_{n-1} term a_3, we have, 4xa_3 4x2.90x10^0 = 1.16x10^1 represented as the same on the machine.
  - **Subtraction**
    Subtracting the machine representation of $4a_3$ from that of $5a_4$ we get 2.40x10^0 which can be accurately represented in the machine as it has three significant digits.
  - **Update**
    Therefore, a_5 will be represented as | 0 | 2 | 4 | 0 | 5 | with a value of 2.40x10^0.

- **n = 5:**
  - **Multiplication**
    As n = 5, a_n is a_5. Therefore the representation of 5xa_n will be 5x2.40x10^0 = 1.20x10^1 which already has 3 significant digits and therefore no chopping takes place.
    Similarly doing the same for the a_{n-1} term a_4, we have, 4xa_4 4x2.80x10^0 = 1.12x10^1 represented as the same on the machine.
  - **Subtraction**
    Subtracting the machine representation of $4a_4$ from that of $5a_5$ we get 8.00x10^{-1} which can be accurately represented in the machine as it has three significant digits.
  - **Update**
    Therefore, a_6 will be represented as | 0 | 8 | 0 | 0 | 4 | with a value of 8.00x10^{-1}.

- **n = 6:**

- **Multiplication**
  As n = 6, a_n is a_6. Therefore the representation of 5xa_n will be 5x8.00x10^{-1} = 4.00x10^0 which already has 3 significant digits and therefore no chopping takes place.
  Similarly doing the same for the a_{n-1} term a_5, we have, 4xa_5 4x2.40x10^0 = 9.60x10^0 represented as the same on the machine.
- **Subtraction**
  Subtracting the machine representation of $4a_5$ from that of $5a_6$ we get -5.60x10^0 which can be accurately represented in the machine as it has three significant digits.
- **Update**
  Therefore, a_7 will be represented as 

with a value of -5.60x10^0.

2 Question 2

2.1 Complexity for $C_{m*l} = A_{m*n} \cdot B_{n*l}$

We know that our final matrix will have $m * l$ elements, so let's look at the number of multiplications and additions required to compute just one element of the final matrix. Let's look at first element , to compute it we need to multiply corresponding elements from first row of A with first column of B. They both have n elements , hence we will have to multiply n times , followed by addition of these n terms (hence, n-1 additions) to get just one element of C. Therefore total multiplications and additions will be total number of elements in C times operations needed for one element , that is:
Number of multiplications: $n * (m * l)$
Number of additions: $(n-1) * (m * l)$
Complexity: = O(mnl)

2.2 $a = u(v^T w) \quad ; u, v, w, a \in R^{n*1}$

We can use the same formula we generated in the above question. First consider the operation $(v^T w)$, here dimentions of v^T are 1 X n and dimensions of w are n X 1, using the above formula we get number of multiplications as $n * 1 * 1$ and number of additions as $(n-1) * 1 * 1$. Finally, this scalar is multiplied with vector u, scalar will be multiplied with all the n elements in u, therefore total operations will be as follows:
Multiplications: n (from $v^T w$) $+n$ (from scalar times u) $= 2n$
Additions: $n - 1$ (only coming from $v^T w$)
Complexity: O(n)

2.3 $b = (uv^T)w \quad ; u, v, w, b \in R^{n*1}$

Again using the same formula we derived in part a. First considering uv^T , here dimensions of u and v are n X 1 , their outer product will result into a matrix of n X n dimensions. Using the formula we derived , number of multiplications for this operation will be $n * 1 * n = n^2$ and number of additions will be 0, this is visualised as each element in u will be multiplied by all the elements in v to produce one element of the resulting n X n matrix. Next this resulting n X n matrix is multiplied by n X 1 vector w. Using the same formula again , number of multiplications will be $n * n * 1 = n^2$ and number of additions will be $n * (n - 1) * 1$. Therefore total operations will be:
Multiplications: $n^2 + n^2 = 2n^2$
Additions: $n * (n - 1)$
Complexity: O(n^2)

2.4 Given $A, B, C \in R^{10000*2}$ compare complexity of $(AB^T)C$ and $A(B^TC)$

$(AB^T)C$**:**
Again using the same formula from the first part:
Number of multiplications in order of operations: $2 * 10000 * 10000 + 10000 * 2 * 10000 = 4 * 10^8$ and
Number of Additions : $(2-1)*10000*10000+(10000-1)*10000*2 = 299980000$
$A(B^TC)$**:**
Following same strategy as last part and caculating number of operations in order as follows:
Number of Multiplications: $10000 * 2 * 2 + 2 * 10000 * 2 = 8 * 10^4$
Number of Additions: $(10000 - 1) * 2 * 2 + (2 - 1) * 10000 * 2 = 59996$
Based on the number of computations we should use $A(B^TC)$ since it has much lesser number of computations required almost by a factor of 10^4 compared to $(AB^T)C$.

3 Question 3

3.1 Analytical Solution:

Given , $\frac{d^2u}{dt^2} = 0$ with $u(0) = 1.64$ and $\left.\frac{du}{dt}\right|_{t=0} = 0$,
Integrate once to get,
$\frac{du}{dt} = c_1$, here c_1 is constant of integration,
Integrate again to get,
$u(t) = c_1t + c_2$ Using initial condition , $u(0) = 1.64$, we get $c_2 = 1.64$, next using $\left.\frac{du}{dt}\right|_{t=0} = 0$ we get $c_1 = 0$, therefore
$u(t) = 1.64$

3.2 Taylor Series Approximation:

To show:

$$\frac{u(t+3\delta t) - 3u(t+\delta t) + 2u(t)}{3(\delta t)^2} = \frac{d^2u}{dt^2} + O(t)$$

Using Taylor Series expression we get,

$$u(t+3\delta t) = u(t) + \frac{du}{dt}\cdot(3\delta t) + \frac{d^2u}{dt^2}\cdot\frac{(3\delta t)^2}{2!} + O((\delta t)^3)$$

$$u(t+\delta t) = u(t) + \frac{du}{dt}\cdot\delta t + \frac{d^2u}{dt^2}\cdot\frac{\delta t^2}{2!} + O((\delta t)^3)$$

Substituting the $u(t+3\delta t)$ and $u(t+\delta t)$ into the expression $\frac{u(t+3\delta t)-3u(t+\delta t)+2u(t)}{3(\delta t)^2}$, we get on simplification:

$$\frac{\frac{6d^2u}{dt^2}\cdot\frac{\delta t^2}{2!} + O((3\delta t)^3) - 3O((\delta t)^3)}{3(\delta t)^2}$$

This simplifies to:

$$\frac{d^2u}{dt^2} + \frac{O((3\delta t)^3) - 3O((\delta t)^3)}{3(\delta t)^2}$$

Now , in the limit δt tends to 0 , the expression $\frac{O((3\delta t)^3)-3O((\delta t)^3)}{3(\delta t)^2}$ will be lesser than $O(\delta t)$ since the higher order terms will decrease exponentially as power of δt increases! Hence it can be approximated to $O(\delta t)$.

3.3 Recurrence Relation:

Given $u_n = u(n\delta t)$
First to discretize let's convert the continuous variable t into time steps as $t = n\delta t$, therefore $3\delta t = 3$ and $\delta t = 1$. Substitute this into the Taylor approximation we obtained:

$$\frac{u_{n+3} - 3u_{n+1} + 2u_n}{3} = \frac{d^2u}{dt^2} + O(1)$$

We are already given that $\frac{d^2u}{dt^2} = 0$ and $O(1)$ corresponds to a finite error between all timesteps. From this we get;

$$u_{n+3} = 3u_{n+1} - 2u_n + 3O(1)$$

Assuming this finite error to be negligible since it arises due to higher power terms we are left with the desired recurrence relation!

$$u_{n+3} = 3u_{n+1} - 2u_n$$

3.4 Analytic Solution for u_n

Given $u_0 = u_1 = u_2 = 1.64 = \alpha(say)$; Starting with n=0:

$$u_3 = 3 * u_1 - 2 * u_0 = 1.64 = \alpha$$

n=1:

$$u_4 = 3 * u_2 - 2 * u_1 = 1.64 = \alpha$$

It is easy to recognize that the pattern continues and $u_n = \alpha = 1.64 \ \forall n$ This clearly matches with our analytic solution as we derived in the first part that $u = 1.64$.

3.5 MATLAB Outputs:

```
a =

  Columns 1 through 6

   1.640000000000000   1.640000000000000   1.640000000000000   1.640000000000000   1.640000000000000   1.640000000000000

  Columns 7 through 12

   1.640000000000000   1.640000000000000   1.639999999999999   1.640000000000000   1.639999999999997   1.640000000000001

  Columns 13 through 18

   1.639999999999992   1.640000000000010   1.639999999999972   1.640000000000048   1.639999999999896   1.640000000000199

  Columns 19 through 24

   1.639999999999592   1.640000000000804   1.639999999998379   1.640000000003229   1.639999999993528   1.640000000012930

  Columns 25 through 30

   1.639999999974124   1.640000000051734   1.639999999896514   1.640000000206954   1.639999999586072   1.640000000827835

  Columns 31 through 36

   1.639999998344308   1.640000003311362   1.639999993377254   1.640000013245468   1.639999973509038   1.640000052981898
```

```
  Columns 31 through 36

   1.639999998344308   1.640000003311362   1.639999993377254   1.640000013245468   1.639999973509038   1.640000052981898

  Columns 37 through 42

   1.639999894036178   1.640000211927616   1.639999576144739   1.640000847710493   1.639998304578983   1.640003390842003

  Columns 43 through 48

   1.639993218315961   1.640013563368044   1.639972873263877   1.640054253472210   1.639891493055543   1.640217013888877

  Columns 49 through 54

   1.639565972222210   1.640868055555543   1.638263888888876   1.643472222222209   1.633055555555542   1.653888888888875

  Columns 55 through 60

   1.612222222222208   1.695555555555541   1.528888888888874   1.862222222222207   1.195555555555540   2.528888888888873
```

The MATLAB outputs start to differ significantly from the analytical and hand solutions we got before. The outputs given on running the below code on MATLAB start to differ significantly as the value of n increases!

```
format long
N = 60;
a(1) = 1.64;
a(2) = 1.64;
a(3) = 1.64
for k=0:N-4
    a(k+4) = 3*a(k+2)-2*a(k+1);
end
a
```

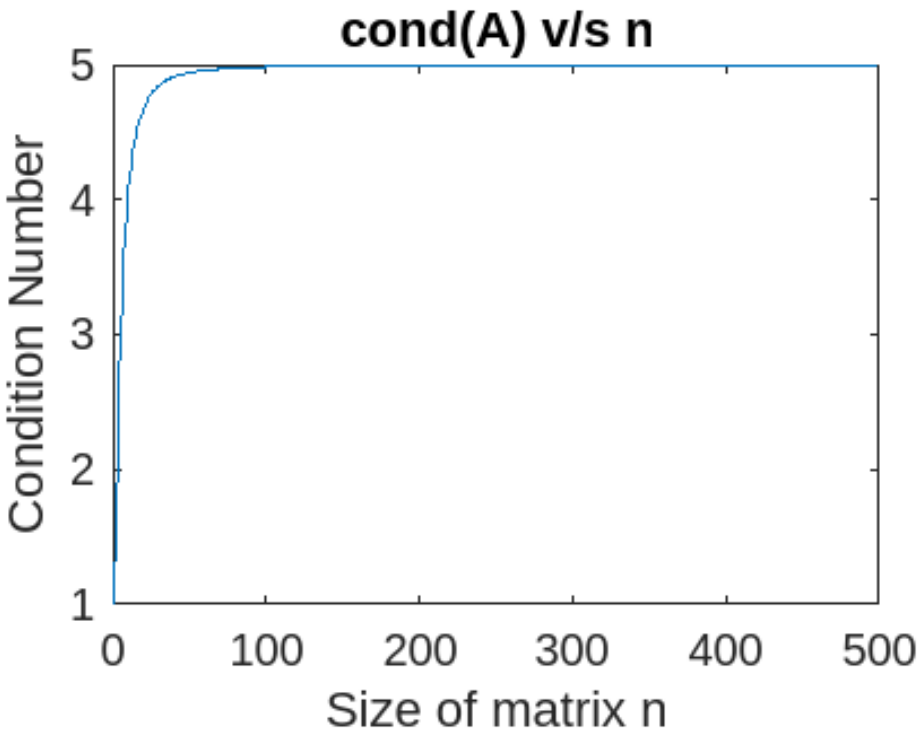
MATLAB uses the IEEE 754 standard to represent numbers which is prone to floating point rounding errors. Since MATLAB has, a limited number of bits available to represent the numbers, it often rounds off the numbers to the nearest representable floating point number. This error keeps accumulating with each recurrence operation leading to huger errors with increase in number of operations.

3.6 Matrix Form

Our Matrix A_{n*n} will look as follows: $\begin{bmatrix} -2 & 3 & 0 & 0 & 0 & \cdots & \\ 0 & -2 & 3 & 0 & 0 & \cdots & \\ 0 & 0 & -2 & 3 & 0 & 0 & \cdots \\ 0 & 0 & 0 & -2 & 3 & 0 & \cdots \end{bmatrix}$ Column Vector b_n will look as: $\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \cdots \\ u_n \end{bmatrix}$ Column Vector u_n will look like: $\begin{bmatrix} u_3 \\ u_4 \\ u_5 \\ \cdots \\ u_{n+3} \end{bmatrix}$

These satisfy $Au = b$ for the recurrence equation we derived above.

3.7 Condition Number Plot:



Condition number exponentially increases with the size of the matrix **n** and saturates at around 5.
NOTE:Relationship between scaling for the condition number and size n of the matrix has been left for the next assignment as soon as it is taught in class!!

4 Question 4

We have to derive the tightest possible upper bound for

$$\frac{|fl(a^T b) - a^T b|}{\|a\|_2 \|b\|_2}$$

in terms of n and μ where n is the size of column vectors a and b and μ represents the machine precision. First let's look at numerator, $|fl(a^T b) - a^T b|$
Let $a^T = [a_1, a_2, a_3, \cdots, a_n]$ and $b^T = [b_1, b_2, b_3, \cdots, b_n]$ and let $s_i = fl(s_{i-1} + a_i * b_i)$, now let's look at floating point error in first term:

$$s_1 = fl(a_1 * b_1) = a_1 * b_1(1 + \delta_x)$$

here δ_x refers to error due to multiplication operator, also we will assume this error is almost same for all multiplication for easier analysis as it doesn't affect the end result! Now let's look at next term:

$$s_2 = fl(s_1 + a_2 * b_2(1 + \delta_x))(1 + \delta_+)$$

, here δ_+ refers to error due to addition operator, $|\delta_x| \leq \mu$ and $|\delta_+| \leq \mu$. For ease of analysis we will again assume $\delta_x = \delta_+ = \delta$. Now expanding the n-th term of s:

$$s_n = a_1 b_1(1 + \delta)^n + a_2 b_2(1 + \delta)^n + a_3 b_3(1 + \delta)^{n-1} + \cdots + a_n b_n(1 + \delta)^2$$

Now using Taylors' approximation for very small values of δ and μ where $|\delta| \leq \mu$ and we have :

$$\prod_{i=1}^{n}(1 + \delta_i) = 1 + \theta_n$$

where $|\theta_n| \leq \frac{nu}{1-nu} = \gamma_n$ this true for most cases as $nu <<< 1$ for most cases! Now rewriting the n-th term:

$$s_n = a_1 b_1(1 + \theta_n) + a_2 b_2(1 + \theta_n) + a_3 b_3(1 + \theta_{n-1} + \cdots + a_n b_n(1 + \theta_2)$$

From this order , we extend to vector notation as:

$$fl(a^T b) = (a + \Delta a)^T b = a^T(b + \Delta b)$$

where, $|\Delta a| \leq \gamma_n |a|$ and $|\Delta b| \leq \gamma_n |b|$. From these, we can finally write the numerator as:

$$|fl(a^T b) - a^T b| \leq \gamma_n |a|^T |b|$$

Next we can use Cauchy -Schwarz inequality $|a^T b| \leq \|a\|\|b\|$ to finally arrive at:

$$\frac{|fl(a^T b) - a^T b|}{\|a\|_2 \|b\|_2} \leq \gamma_n = \frac{nu}{1 - nu}$$

. This can be further strengthened based on conditions over nu based on the implementation of the inner product algorithm, which is followed!