Assignment 1
Sahil Goyal
2020328

1 Absolute Condition Number = $\frac{\text{absolute forward error}}{\text{absolute backward error}}$

$$= \frac{|x - x_0|}{|b - a x_0|} = \frac{|x - x_0|}{|a(x - x_0)|} = \frac{1}{|a|}$$

Relative Condition Number = $\frac{\text{Relative forward error}}{\text{Relative backward error}}$

$$= \frac{\frac{|x - x_0|}{|x_0|}}{\frac{|b - a x_0|}{|a x_0|}}$$

$$= \frac{|x - x_0| \, |a x_0|}{|x_0| \, |a(x - x_0)|}$$

$$= |a|$$

∴ Absolute Condition number is $1/|a|$ and relative Condition number is $|a|$

2

a) $(x-1)^\alpha$

absolute condition number

$$= |f'(x)|$$

$$= \left| \frac{d(x-1)^\alpha}{dx} \right|$$

$$= |\alpha(x-1)^{\alpha-1}|$$

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

$$= \left| \frac{x(\alpha(x-1)^{\alpha-1})}{(x-1)^\alpha} \right|$$

$$= \left| \frac{\alpha x}{x-1} \right|$$

This is large in $\begin{cases} \infty & x = 1 \\ >> 1 & \text{in neighborhood of } x = 1 \\ \text{finite} & \text{otherwise} \end{cases}$

b) $\ln x$

Absolute Condition Number $= |f'(x)|$

$= \left|\frac{d(\ln x)}{dx}\right|$

$= \left|\frac{1}{x}\right|$

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

$= \left|\frac{x(1/x)}{\ln(x)}\right|$

$= \left|\frac{1}{\ln x}\right|$

This is Large in

- ∞ at $x = 1$
- $>> 1$ in neighborhood of $x = 1$
- > 1 for $1 < x < e$
- finite otherwise

Absolute condition number is large in

- ∞ at $x = 0$
- $>> 1$ in neighborhood at $x = 0$
- > 1 $-1 < x < 1$
- finite otherwise

c) $x^{-1}e^x$

Absolute Condition Number $= |f'(x)|$

$$= \left| \frac{d(x^{-1}e^x)}{dx} \right|$$

$$= \left| e^x \left(\frac{1}{x} - \frac{1}{x^2} \right) \right|$$

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

$$= \left| \frac{x \left(e^x \left(\frac{1}{x} - \frac{1}{x^2} \right) \right)}{x^{-1} e^x} \right|$$

$$= |x - 1|$$

this is large in $\begin{cases} \bullet \\ \gg 1 \text{ at } |x| \to \infty \\ \text{finite otherwise} \end{cases}$

Absolute Condition number is large in $\begin{cases} \infty \text{ at } x = 0 \\ \gg 1 \text{ in neighborhood of } x = 0 \\ \gg 1 \text{ at } x \to \infty \\ \text{finite otherwise} \end{cases}$

d) $\frac{1}{(1+x^{-1})}$

Absolute Condition Number $= |f'(x)|$

$$= \left| \frac{d}{dx}\left(\frac{1}{(1+x^{-1})}\right) \right|$$

$$= \frac{1}{(1+x)^2}$$

This is large at
- ∞ at $x = -1$
- $\gg 1$ in neighborhood of $x = -1$
- > 1 $-2 < x \leq 0$
- finite otherwise

Relative Condition No $= \left| \frac{x f'(x)}{f(x)} \right|$

$$= \left| \frac{x\left(1/(1+x)^2\right)}{1/(1+x^{-1})} \right|$$

$$= \left| \frac{1}{1+x} \right|$$

this is large in
- ∞ at $x = -1$
- $\gg 1$ in neighborhood of $x = -1$
- > 1 for $-2 < x < 0$
- finite otherwise

3

a) We are given that

$$f(x(\epsilon)) + \epsilon p(x(\epsilon)) = 0$$

$$x(0) = x^{(0)}$$

We need to prove

$$\left.\frac{dx}{d\epsilon}\right|_{\epsilon=0} = -\frac{p(x^{(0)})}{f'(x^{(0)})}$$

Proof

$$f(x(\epsilon)) + \epsilon p(x(\epsilon)) = 0$$

differentiate wrt ϵ

$$f'(x(\epsilon))\, x'(\epsilon) + p(x(\epsilon)) + \epsilon p'(x(\epsilon))\, x'(\epsilon) = 0$$

(Using chain rule)

We need to find $\left.\frac{dx}{d\epsilon}\right|_{\epsilon=0}$

∴ We set $\varepsilon = 0$

$$\therefore f'(x(0))x'(0) + p(x(0)) + \varepsilon(0)(p(x'(0))x'(0) = 0$$

$$\therefore f'(x(0))x'(0) + p(x(0)) = 0$$

$$\therefore f'(x^*)x'(0) + p(x^*) = 0$$

$$\therefore x'(0) = -\frac{p(x^*)}{f'(x^*)}$$

We have $x(0) = x^*$

$$\therefore x'(0) = \left.\frac{dx}{d\varepsilon}\right|_{\varepsilon=0}$$

$$\therefore \left.\frac{dx}{d\varepsilon}\right|_{\varepsilon=0} = -\frac{p(x^*)}{f'(x^*)}$$

∴ Thus Proved

b) $f(x) = (x-1)(x-2) \ldots\ldots (x-20)$

$\therefore f(x)$ can be written as

$f(x) = a_0 + a_1 x + a_2 x^2 \ldots a_{20} x^{20}$

We have expressed a_{19} inaccurately

$\therefore$ Let $p(x) = x^{19}$

$\therefore$ We will use the formula

$$\left.\frac{dx}{d\epsilon}\right|_{\epsilon=0} = -\frac{p(x^j)}{f'(x^j)}$$

to show root competation for $x^j = j$

Differentiate $f(x)$ wrt x

$$f'(x) = (x-2)(x-3)\ldots(x-20)$$

$$+(x-1)(x-3)(x-4)\ldots(x-20)$$

$$+(x-1)\ldots(x-2)(x-4)\ldots(x-20)$$

$$\vdots$$

$$(x-1)(x-2)\ldots\ldots(x-19)$$

∴ We can clearly see that this pattern continues for j ranging from 1 to 20

For a particular value of j, the 19^{q} terms that contain $x-j$ will turn into 0, while the 0-term that does not contain $x-j$ remains the same

$$\therefore f'(x) = (j-1)(j-2)\ldots(j-k)$$

(It is given in question that $x = j$)

$$\therefore \quad f'(x^0) = \prod_{\substack{k=1 \\ k \neq j}}^{20} (j-k)$$

$$\left.\frac{dx}{d\xi}\right|_{\xi=0,\, x^0=j} = -\frac{p(x^0)}{f'(x)}$$

$$= -\frac{p(j^{19})}{f'(j^{19})}$$

$$= \frac{-j^{19}}{\prod_{\substack{k=1 \\ k \neq j}}^{20} (j-k)}$$

$$\left.\frac{dx}{d\xi}\right|_{\substack{\xi=0 \\ x^0=j}} = -1^T \prod_{\substack{k=1 \\ j \neq k}} \left(\frac{j}{j-k}\right)$$

$\therefore$ Thus Proved

2) For $x^{(0)} = 1$

$$\left.\frac{dx}{d\varepsilon}\right|_{\substack{\varepsilon=0 \\ x^{(0)}=1}}$$

$$= \frac{-(1)^{19}}{(1-1)(1-2)(1-3)\ldots(1-20)}$$

$$= \frac{-1}{(-1)(-2)\ldots(-19)}$$

$$= \frac{1}{19!}$$

∴ We can see that $dx/d\varepsilon|_{\substack{\varepsilon=0 \\ x^{(0)}=1}}$

is $1/19!$ which shows that for change in ε, change in x is very small.

∴ We can see that root is stable

For $x^0 = 20$

$$\left.\frac{dx}{d\varepsilon}\right|_{\substack{\varepsilon=0 \\ x^0=20}} = \frac{-(20)^{19}}{(20-1)(20-2)\cdots(20-19)}$$

$$= \frac{-(20)^{19}}{19!}$$

∴ We can see that the magnitude of $\left.\frac{dx}{d\varepsilon}\right|_{\substack{\varepsilon=0 \\ x^0=20}}$ is $\frac{20^{19}}{19!}$,

It is very large,

∴ small changes in the input will lead to large changes in the output.

∴ We can see that the root is unstable

∴ $x^0 = 1$ is more stable to this perturbation

4

a) This code snippet ~~keeps on~~ assigns a value 1 to a variable a and keeps on halving it, and printing the value of a. We can see that 5e-324 is the last number printed before 0. It is approximately equal to the UFL (Underflow) in the IEEE double precision system (754 standard).

b) This code snippet prints out 1.1102230246251565e-16

This is then number which when added to 1 does not change its value in floating point arithmetic. This number is the half of ε_m (Machine epsilon) which is the smallest number which when added to 1 gives a number greater than 1 in floating point arithmetic. As eps is smaller than ε_m, it will not change the number when added to 1

c) This code snippet keeps on doubling the numbers starting from 1 and printing the value of a. We can see that the last number printed is inf which is approximately equal to the OFL (overflow) of the IEEE double precision system (754 standard)

5 Analysis:

Ideally,

$\tan(x + 2n\pi) = \tan(x)$

where n is any integer

But, we have observed that as the value of j increases, the error increases and hence the condition number also increases.

We can see that for $j = 0$, the condition number is 14.137166941154069O while for $j = 20$, the condition number is 1258173865379857956864.00 00000000

This is because, as x increases, the rounding error also increases.

Because of this, small changes in input can cause large changes in the output (value of tan x)

Q5:

Output is:

```
The value of j is 0
(x,tan(x) = (7.0685834705770345,0.9999999999999994)
Condition number is 14.1371669411540690
Error in the input is 2.7755575615628914e-16
The value of j is 1
(x,tan(x) = (63.6172512351933079,0.9999999999999897)
Condition number is 127.2345024703866159
Error in the input is 5.162537064506978e-15
The value of j is 2
(x,tan(x) = (629.1039288813560688,0.9999999999999456)
Condition number is 1258.2078577627121376
Error in the input is 2.7200464103316335e-14
The value of j is 3
(x,tan(x) = (6283.9707053429829102,0.9999999999979700)
Condition number is 12567.9414106859658204
Error in the input is 1.0150214002635494e-12
The value of j is 4
(x,tan(x) = (62832.6384699592599645,0.9999999999954939)
Condition number is 125665.2769399185199291
Error in the input is 2.2530310950230614e-12
The value of j is 5
(x,tan(x) = (628319.3161161219468340,0.9999999998033864)
Condition number is 1256638.6322322436608374
Error in the input is 9.830680713918129e-11
The value of j is 6
(x,tan(x) = (6283186.0925777498632669,0.9999999999777867)
Condition number is 12566372.1851554978638887
Error in the input is 1.110667113835007e-11
The value of j is 7
(x,tan(x) = (62831853.8571940287947655,1.0000000012561285)
Condition number is 125663707.7143880575895309
Error in the input is 6.280642672606973e-10
The value of j is 8
(x,tan(x) = (628318531.5033566951751709,0.9999997681704147)
Condition number is 1256637063.0067470073699951
Error in the input is 1.1591479265326798e-07
The value of j is 9
(x,tan(x) = (6283185307.9649848937988281,1.0000005069523146)
Condition number is 12566370615.9315834045410156
Error in the input is 2.534761572858487e-07
The value of j is 10
(x,tan(x) = (62831853072.5812606811523438,0.9999954970141940)
Condition number is 125663706146.4365539550781250
Error in the input is 2.2514929029820325e-06
The value of j is 11
```

```
(x,tan(x) = (628318530718.7440185546875000,0.9999453990134111)
Condition number is 1256637063310.7761230468750000
Error in the input is 2.7300493253771445e-05
The value of j is 12
(x,tan(x) = (6283185307180.3710937500000000,0.9984385423410914)
Condition number is 12566385957667.0507812500000000
Error in the input is 0.0007807278761999814
The value of j is 13
(x,tan(x) = (62831853071796.6484375000000000,0.9965461389148867)
Condition number is 125664458272343.4218750000000000
Error in the input is 0.0017269202065067328
The value of j is 14
(x,tan(x) = (628318530717959.3750000000000000,0.8900802593986616)
Condition number is 1265166139489326.2500000000000000
Error in the input is 0.05458936004990577
The value of j is 15
(x,tan(x) = (6283185307179587.0000000000000000,0.5766517306609554)
Condition number is 14519188958191302.0000000000000000
Error in the input is 0.1832041468287601
The value of j is 16
(x,tan(x) = (62831853071795864.0000000000000000,-0.9682197486526081)
Condition number is 125729248279028768.0000000000000000
Error in the input is 0.983596861932204
The value of j is 17
(x,tan(x) = (628318530717958656.0000000000000000,-2.0518446488895292)
Condition number is 1595433312584221184.0000000000000000
Error in the input is 1.2018869924834348
The value of j is 18
(x,tan(x) = (6283185307179586560.0000000000000000,5.5747653335805616)
Condition number is 36154359831181299712.0000000000000000
Error in the input is 0.7950382322343758
The value of j is 19
(x,tan(x) = (62831853071795863552.0000000000000000,-8.7118803024535953)
Condition number is 554595786581180678144.0000000000000000
Error in the input is 1.1002886263819651
The value of j is 20
(x,tan(x) = (628318530717958668288.0000000000000000,-1.0506941034214516)
Condition number is 1258173865379857956864.0000000000000000
Error in the input is 1.024094635461799
```

6

a) We need to prove that $\|x\|_A := \|Ax\|$ is a vector norm on R^n

∴ We need to prove

i) $\|x\|_A \geq 0$ and $\|x\|_A = 0$ iff $x = 0$

ii) $\|\alpha x\|_A = |\alpha| \|x\|_A \quad \forall \alpha \in R, x \in R^n$

iii) $\|x+y\|_A \leq \|x\|_A + \|y\|_A \quad \forall y, x \in R^n$

Proof:

i) $\|x\|_A = \|Ax\|$

∴ A is a full rank matrix of dimensions $n \times n$, and x is a vector of dimensions $n \times 1$

(Ax is a vector norm, so Ax must be vector ∴ $x \in R^n$)

∴ Ax is a vector dimensions $n \times 1$

∴ Let $b = Ax$

where b is a vector of dimensions $n \times 1$

$$\|x\|_A = \|Ax\|$$
$$= \|b\|$$

∴ Norm of a vector b is

$$\|b\|_p = \left(\sum_{i=1}^{m} |b_i|^p\right)^{1/p}$$

∴ We can see that as we are using the summation of absolute values of the vector raised to p, they must be non negative.

∴ The norm of b will always be greater or equal to 0

When b is a zero vector, all elements of b will be 0 and norm of b will be 0

∴ For b to be a zero vector, Ax must be a zero vector. But this is only possible when $x = 0$, as A is a full rank matrix.

Because of this, the only solution to $Ax = 0$ is $x = 0$

(Product of vector with full rank matrix cannot be zero vector unless original vector is also 0 vector)

$\therefore \|x\|_A \geq 0$ and $\|x\|_A = 0$ iff $x = 0$

$\therefore$ Thus Proved

ii) $\|\alpha x\|_A$

$= \|\alpha A x\|$

$= \|\alpha b\|$

$= \left(\sum_{i=1}^{m} (|b_i|)^p\right)^{1/p}$

$= |\alpha| \left(\sum_{i=1}^{m} |b_i|^p\right)^{1/p}$

$= |\alpha| \|b\|$

$= |\alpha| \|Ax\|$

$= |\alpha| \|x\|_A$

$\therefore \|\alpha x\|_A = |\alpha| \|x\|_A$

$\therefore$ Thus Proved

(iii) $\|x+y\|_A$

$= \|A(x+y)\|$

$= \|Ax+Ay\|$

$\leq \|Ax\| + \|Ay\|$ (Property of $\|\cdot\|$ Norms)

$\|x+y\|_A \leq \|x\|_A + \|y\|_A$

∴ Thus Proved

We have proved all three properties for $\|x\|_A$

$\|x\|_A$ is a vector norm on R^n

∴ Thus Proved

b) We need to prove that the Frobenius norm satisfies the three properties of norms.

Frobenius norm is

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}$$

The three properties are

i) $\|A\|_F \geq 0$ and $\|A\|_F = 0$ iff $A = 0$

ii) $\|\alpha A\|_F = |\alpha| \|A\|_F \quad \forall \alpha \in \mathbb{R}, A \in \mathbb{R}^{m \times n}$

iii) $\|A+B\|_F \leq \|A\|_F + \|B\|_F \quad \forall A, B \in \mathbb{R}^{m \times n}$

Proof:

i) $\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}$

As we are taking modulus of every element it is clear that $\|A\|_F$ will always be greater than or equal to 0.

The only case where $\|A\|_F$ will be equal to 0 will be when every element of A is 0, i.e A is zero matrix

$\therefore \|A\|_F \geq 0$ and $\|A\|_F = 0$ if $A = 0$

ii)

$\|\alpha A\|_F$

$$= \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |\alpha a_{ij}|^2 \right)^{1/2}$$

$$= \left(\alpha^2 \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

$$= |\alpha| \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

$$= |\alpha| \, \|A\|_F$$

$\therefore \|\alpha A\|_F = |\alpha| \, \|A\|_F$

iii) $\|A+B\|_F$

Here, we will use the Cauchy Schwarz inequality

$$\sum_{i=1}^{n} a_i b_i \leq \left(\sum_{i=1}^{n} a_i^2\right)^{1/2} \left(\sum_{i=1}^{n} y_i^2\right)^{1/2}$$

$$\|A+B\|_F$$

$$= \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}+b_{ij}|^2\right)^{1/2}$$

Take square on both sides

$$\|A+B\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}+b_{ij}|^2$$

$$\|A+B\|_F^2 \leq \sum_{i=1}^{m}\sum_{j=1}^{n} (|a_{ij}|+|b_{ij}|)^2$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \left(|a_{ij}|^2 + 2|a_{ij}||b_{ij}| + |b_{ij}|^2\right)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} |b_{ij}|^2 + \sum_{i=1}^{m}\sum_{j=1}^{n} 2|a_{ij}||b_{ij}|$$

$$\leq \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 + \sum_{i=1}^{m}\sum_{j=1}^{n} |b_{ij}|^2 +$$

$$+ 2\left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} |b_{ij}|^2\right)$$

(Cauchy Schwarz inequality)

$$= \left(\left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2} + \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |b_{ij}|^2\right)^{1/2}\right)^2$$

$$= \left(\|A\|_F + \|B\|_F\right)^2$$

$$\therefore \|A+B\|_F^2 \leq \left(\|A\|_F + \|B\|_F\right)^2$$

$$\therefore \|A+B\|_F \leq \|A\|_F + \|B\|_F$$

∴ Thus Proved

∴ Frobenius Norm satisfies the 3 properties of a norm.

c) $E = uv^T$

where $u \in R^m$, $v \in R^n$

$\therefore E \in R^{m \times n}$

We need to show that $\|E\|_2 = \|E\|_F = \|u\|_2 \|v\|_2$

Proof:

$$\|E\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2} = \|E\|_2$$

$$= \sqrt{\text{trace}(EE^T)}$$

(Trace of a matrix is sum of diagonal elements)

$$= \sqrt{\text{tr}((uv^T)(uv^T)^T}$$

$$= \sqrt{\text{tr}(uv^T v u^T)}$$

$$= \sqrt{\text{tr}(u^T u \cdot v^T v)}$$

(Trace of matrix is associative over product)

$$= \sqrt{\text{tr}(u^T u)\,\text{tr}(v^T v)}$$

(Trace of $R^{1 \times 1}$ matrix is the element itself of the matrix)

We can split the trace into 2

$$\sqrt{\text{tr}(u^T u)}\ \sqrt{\text{tr}(v^T v)}$$

$$= \|u\|_2 \|v\|_2$$

$$\therefore \|E\|_f = \|F\|_2 = \|u\|_2 \|v\|_2$$

Thus Proved

7 We can see from output of code that second formula converges faster than the first.

We can see this because by the 5000 th iteration first formula reaches till 0.577196615681 66 while the second formula reaches 0.577215664567832 7

∴ Thus Proved

Q7:

Output for first equation is:

```
Using equation 6.1
current n is 100 and the value of term is 0.5822073316515288
current n is 200 and the value of term is 0.5797135815734098
current n is 300 and the value of term is 0.5788814056433012
current n is 400 and the value of term is 0.5784651440685238
current n is 500 and the value of term is 0.5782153315683285
current n is 600 and the value of term is 0.5780487667534508
current n is 700 and the value of term is 0.5779297805478292
current n is 800 and the value of term is 0.5778405346932214
current n is 900 and the value of term is 0.577771117576444
current n is 1000 and the value of term is 0.5777155815682065
current n is 1100 and the value of term is 0.5776701414855578
current n is 1200 and the value of term is 0.5776322736978301
current n is 1300 and the value of term is 0.5776002309764809
current n is 1400 and the value of term is 0.5775727652416682
current n is 1500 and the value of term is 0.5775489611978291
current n is 1600 and the value of term is 0.5775281323494514
current n is 1700 and the value of term is 0.5775097537135299
current n is 1800 and the value of term is 0.5774934169591495
current n is 1900 and the value of term is 0.5774787997122512
current n is 2000 and the value of term is 0.5774656440682016
current n is 2100 and the value of term is 0.577453741243179
current n is 2200 and the value of term is 0.5774429204111771
current n is 2300 and the value of term is 0.5774330404528918
current n is 2400 and the value of term is 0.5774239837672805
current n is 2500 and the value of term is 0.5774156515681996
current n is 2600 and the value of term is 0.5774079602664202
current n is 2700 and the value of term is 0.5774008386555254
current n is 2800 and the value of term is 0.5773942257008384
current n is 2900 and the value of term is 0.5773880687857824
current n is 3000 and the value of term is 0.5773823223089227
current n is 3100 and the value of term is 0.5773769465525795
current n is 3200 and the value of term is 0.5773719067634975
current n is 3300 and the value of term is 0.5773671724007521
current n is 3400 and the value of term is 0.5773627165162711
current n is 3500 and the value of term is 0.5773585152416505
current n is 3600 and the value of term is 0.5773545473603523
current n is 3700 and the value of term is 0.5773507939494777
current n is 3800 and the value of term is 0.5773472380778735
current n is 3900 and the value of term is 0.5773438645508655
current n is 4000 and the value of term is 0.5773406596931707
current n is 4100 and the value of term is 0.5773376111636459
current n is 4200 and the value of term is 0.5773347077964406
current n is 4300 and the value of term is 0.577331939464333
current n is 4400 and the value of term is 0.5773292969607322
```

```
current n is 4500 and the value of term is 0.5773267718973916
current n is 4600 and the value of term is 0.5773243566154296
current n is 4700 and the value of term is 0.5773220441077846
current n is 4800 and the value of term is 0.5773198279512748
current n is 4900 and the value of term is 0.5773177022470506
current n is 5000 and the value of term is 0.577315661568166
```

Output of second equation is:

```
Using equation 6.2
current n is 100 and the value of term is 0.5772197901404903
current n is 200 and the value of term is 0.5772167013748222
current n is 300 and the value of term is 0.5772161263242399
current n is 400 and the value of term is 0.5772159246680912
current n is 500 and the value of term is 0.5772158312352449
current n is 600 and the value of term is 0.577215780449559
current n is 700 and the value of term is 0.5772157498141715
current n is 800 and the value of term is 0.5772157299243794
current n is 900 and the value of term is 0.5772157162847442
current n is 1000 and the value of term is 0.5772157065265553
current n is 1100 and the value of term is 0.5772156993055031
current n is 1200 and the value of term is 0.5772156938126143
current n is 1300 and the value of term is 0.577215689537403
current n is 1400 and the value of term is 0.5772156861448545
current n is 1500 and the value of term is 0.5772156834077089
current n is 1600 and the value of term is 0.5772156811674058
current n is 1700 and the value of term is 0.5772156793105871
current n is 1800 and the value of term is 0.5772156777544755
current n is 1900 and the value of term is 0.5772156764374801
current n is 2000 and the value of term is 0.5772156753129938
current n is 2100 and the value of term is 0.5772156743452568
current n is 2200 and the value of term is 0.577215673506438
current n is 2300 and the value of term is 0.5772156727746101
current n is 2400 and the value of term is 0.5772156721323229
current n is 2500 and the value of term is 0.5772156715655328
current n is 2600 and the value of term is 0.5772156710628664
current n is 2700 and the value of term is 0.5772156706149998
current n is 2800 and the value of term is 0.5772156702142466
current n is 2900 and the value of term is 0.5772156698542288
current n is 3000 and the value of term is 0.5772156695296005
current n is 3100 and the value of term is 0.5772156692358834
current n is 3200 and the value of term is 0.5772156689692576
current n is 3300 and the value of term is 0.5772156687264989
current n is 3400 and the value of term is 0.5772156685048309
current n is 3500 and the value of term is 0.5772156683019034
current n is 3600 and the value of term is 0.5772156681156311
current n is 3700 and the value of term is 0.577215667944273
```

```
current n is 3800 and the value of term is 0.5772156677862554
current n is 3900 and the value of term is 0.5772156676402354
current n is 4000 and the value of term is 0.5772156675050191
current n is 4100 and the value of term is 0.5772156673795781
current n is 4200 and the value of term is 0.5772156672629993
current n is 4300 and the value of term is 0.5772156671544533
current n is 4400 and the value of term is 0.5772156670532187
current n is 4500 and the value of term is 0.5772156669586632
current n is 4600 and the value of term is 0.5772156668702006
current n is 4700 and the value of term is 0.5772156667873283
current n is 4800 and the value of term is 0.5772156667095789
current n is 4900 and the value of term is 0.5772156666365351
current n is 5000 and the value of term is 0.5772156665678327
```

8.

a) Explanation for Random matrix:

As we can see, the relative error in solution of random matrices is very small. We can also see that on average the error in random matrices decreases when we use partial pivoting compared to when we dont use pivoting.

For Hilbert Matrix:

They have very large condition numbers, as it has many rounding errors because its elements are of the form $1/(i+j-1)$ which need very high precision. The high condition number explains the high error in solution. Pivoting is needed because the values of the entries decrease, and there are large differences between the values in different rows and columns.

We can see that error in Hilbert matrix with pivoting is less compared to error without pivoting

For ones, minus ones matrix

They have very small condition number, as the matrix only contains 1's and -1's and error is 0. ~~transpose~~

Pivoting is not needed here as the solution is already extremely accurate. The error coming out to be 0 both when we do and do not apply pivoting.

Q8)

c.

Output for Gaussian elimination with no pivoting:

```
For n =  10
For Random Matrix:
Condition Number =
528.0691818994849
Error from unpivoted solve =
1.7134881620890917e-15
Residual from unpivoted solve =
8.994789908602516e-16
Error from solve =
2.9486445059727435e-14
Residual from solve =
5.578331894996019e-16
For Hilbert Matrix:
Condition Number =
16024930538618.01
Error from unpivoted solve =
4.5939517967567676e-05
Residual from unpivoted solve =
9.010594080615092e-16
Error from solve =
3.4674419835843716e-05
Residual from solve =
8.793439420581891e-16
For one,minus one Matrix:
Condition Number =
6.313751514675045
Error from unpivoted solve =
0.0
Residual from unpivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  20
For Random Matrix:
Condition Number =
82.60408874880953
Error from unpivoted solve =
1.3809406345469375e-13
Residual from unpivoted solve =
1.1525931936519502e-13
Error from solve =
1.2798902193149253e-15
```

```
Residual from solve =
2.3761678997493533e-15
For Hilbert Matrix:
Condition Number =
6.806966466008104e+18
Error from unpivoted solve =
3.150016656798906
Residual from unpivoted solve =
2.570495205447063e-15
Error from solve =
3.590213258628455
Residual from solve =
1.8693106197088372e-15
For one,minus one Matrix:
Condition Number =
12.706204736174707
Error from unpivoted solve =
0.0
Residual from unpivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  30
For Random Matrix:
Condition Number =
460.56868410308954
Error from unpivoted solve =
2.976107390805341e-14
Residual from unpivoted solve =
4.494345410317162e-14
Error from solve =
5.004877514095219e-15
Residual from solve =
5.665124776577122e-15
For Hilbert Matrix:
Condition Number =
4.728514165461419e+19
Error from unpivoted solve =
55.30112699331478
Residual from unpivoted solve =
9.813091785407791e-15
Error from solve =
42.05721893278598
Residual from solve =
1.0852757259154574e-14
For one,minus one Matrix:
```

```
Condition Number =
19.081136687728225
Error from unpivoted solve =
0.0
Residual from unpivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  40
For Random Matrix:
Condition Number =
301.82650926454505
Error from unpivoted solve =
1.0584544971386907e-12
Residual from unpivoted solve =
5.757295329524611e-13
Error from solve =
2.9966768378143897e-15
Residual from solve =
4.179751905981344e-15
For Hilbert Matrix:
Condition Number =
5.697452796254398e+18
Error from unpivoted solve =
48.42902059618393
Residual from unpivoted solve =
2.2541690868016152e-14
Error from solve =
7.506357693270973
Residual from solve =
5.263860016772637e-15
For one,minus one Matrix:
Condition Number =
25.45169957935708
Error from unpivoted solve =
0.0
Residual from unpivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
```

For Gaussian elimination with partial pivoting:

```
For n =  10
```

```
For Random Matrix:
Condition Number =
90.01866595272816
Error from pivoted solve =
1.6612587315373646e-15
Residual from pivoted solve =
2.2834969560822236e-15
Error from solve =
1.5519313616435815e-15
Residual from solve =
7.806858883768762e-16
For Hilbert Matrix:
Condition Number =
16024930538618.01
Error from pivoted solve =
0.00012431167596888245
Residual from pivoted solve =
1.3480087372939849e-15
Error from solve =
3.4674419835843716e-05
Residual from solve =
8.793439420581891e-16
For one,minus one Matrix:
Condition Number =
6.313751514675045
Error from pivoted solve =
0.0
Residual from pivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  20
For Random Matrix:
Condition Number =
254.23198829881542
Error from pivoted solve =
3.4204059704919426e-15
Residual from pivoted solve =
3.8567749967776725e-15
Error from solve =
5.3357161478015186e-15
Residual from solve =
2.606953250304646e-15
For Hilbert Matrix:
Condition Number =
6.806966466008104e+18
```

```
Error from pivoted solve =
44.49645652235686
Residual from pivoted solve =
1.309730709377927e-14
Error from solve =
3.590213258628455
Residual from solve =
1.8693106197088372e-15
For one,minus one Matrix:
Condition Number =
12.706204736174707
Error from pivoted solve =
0.0
Residual from pivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  30
For Random Matrix:
Condition Number =
379.36399378833136
Error from pivoted solve =
1.1938675117717591e-14
Residual from pivoted solve =
3.748505456350477e-15
Error from solve =
9.778348771066036e-15
Residual from solve =
3.529101202518859e-15
For Hilbert Matrix:
Condition Number =
4.728514165461419e+19
Error from pivoted solve =
58.465492383865254
Residual from pivoted solve =
2.48680920304251e-14
Error from solve =
42.05721893278598
Residual from solve =
1.0852757259154574e-14
For one,minus one Matrix:
Condition Number =
19.081136687728225
Error from pivoted solve =
0.0
Residual from pivoted solve =
```

```
0.0
Error from solve =
0.0
Residual from solve =
0.0
For n =  40
For Random Matrix:
Condition Number =
11664.150218066532
Error from pivoted solve =
1.7485889543637915e-13
Residual from pivoted solve =
7.465948721503936e-15
Error from solve =
3.617117859564199e-13
Residual from solve =
5.631550408650825e-15
For Hilbert Matrix:
Condition Number =
5.697452796254398e+18
Error from pivoted solve =
20.2496214518533
Residual from pivoted solve =
1.2184416607749135e-14
Error from solve =
7.506357693270973
Residual from solve =
5.263860016772637e-15
For one,minus one Matrix:
Condition Number =
25.45169957935708
Error from pivoted solve =
0.0
Residual from pivoted solve =
0.0
Error from solve =
0.0
Residual from solve =
0.0
```