SCIENTIFIC COMPUTING - MTH 373/573
HOMEWORK 1

Ans-1

Given :- Absolute Forward Error = | x - x0 | and Absolute Backward Error = | b-ax | = | a(x0-x) |

So, Absolute Condition Number = Absolute Forward Error / Absolute Backward Error

= | x-x0 | / | -a(x-x0) |

= | x-x0 | / | a(x-x0) | = | x-x0| / |a|*| x-x0 |

= 1/|a|

Now, Relative Forward Error = Absolute Forward Error / x0 and Relative Backward Error = Absolute Backward Error / b

So, Relative Condition Number = Relative Forward Error / Relative Backward Error

= | x - x0 | / |x0| / | b-ax | / |b|

= | x - x0 | * |b| / |x0| * | b-ax |

= | x-x0 | * |a|*|x0| / |x0| * | ax0-ax |

= | x - x0 | * |a| / |a| * | x - x0 |

= 1

Ans-2

Ans ② (a) $(x-1)^a = f(x)$

$\Rightarrow$ Absolute Condition Number $= \left|\frac{d((x-1)^a)}{d(x)}\right| = |f'(x)|$

$= |a(x-1)^{a-1}|$

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

$= \left|\frac{x \times a(x-1)^{a-1}}{(x-1)^a}\right| = \left|\frac{ax}{x-1}\right|$

$\Rightarrow$ Now if $a<1$ then abs. condn number will be large if $x \to 1$,

$\Rightarrow$ So, for $a>1$ absolute condition number if $x \to \pm\infty$ then it will be large and for relative condition number if $x \to 1$ then it will be large.

(b) $\ln x = f(x)$

$\Rightarrow$ Absolute Condition Number $= \left|\frac{d(f(x))}{d(x)}\right| = \left|\frac{1}{x}\right|$

$\Rightarrow$ Relative Condition Number $= \left|\frac{xf'(x)}{f(x)}\right| = \left|\frac{x \times \frac{1}{x}}{\ln x}\right|$

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

$\Rightarrow$ So, for absolute condn number if $x \to 0$ then it will be large and for relative condn number number if $x \to 1$ then it will be large.

(c) $x^{-1}e^{x} = f(x)$

⇒ Absolute Cond^n Number $= \left|\dfrac{d(f(x))}{dx}\right| = \left|\dfrac{d\left(\frac{e^x}{x}\right)}{d(x)}\right|$

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

⇒ Relative Cond^n Number $= \left|\dfrac{x \times f'(x)}{f(x)}\right| = \left|\dfrac{\cancel{x} \times \cancel{e^x}(x-1)/\cancel{x^2}}{\cancel{e^x}/\cancel{x}}\right|$

$$= |x-1|$$

⇒ So, for absolute cond^n number if $x \to 0$ then it will be large and for relative cond^n number if $x \to \pm\infty$ then it will be large.

(d) $\dfrac{1}{1+x^{-1}} = f(x)$

⇒ Absolute cond^n number $= \left|\dfrac{d(f(x))}{d(x)}\right| = \left|\dfrac{d\left(\frac{x}{1+x}\right)}{d(x)}\right|$

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

⇒ Relative cond^n number $= \left|\dfrac{x f'(x)}{f(x)}\right| = \left|\dfrac{\cancel{x} \times \frac{1}{(1+x)^{\cancel{2}}}}{\frac{\cancel{x}}{\cancel{1+x}}}\right|$

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

⇒ So, for absolute cond^n number if $x \to -1$ then it will be large and for relative cond^n number if $x \to -1$ then it will be large.

Ans-3 [Note : We are denoting ε as e]

(a) Given :- f(x(e)) + e(p(x(e)) = 0 ……. (1)
and, x(0) = x*
To show :- dx/de (at e = 0) = -p(x*) / f'(x*)
So, differentiating equation (1) with respect to e, i.e.,
=> f'(x(e)) * dx/de + p(x(e)) + e(p'(x(e)) * dx/de = 0
=> dx/de * (f'(x(e)) + e*p'(x(e))) + p(x(e)) = 0
=> dx/de = - p(x(e)) / f'(x(e)) + e*p'(x(e))
So according to question, we have to calculate the value of differentiated equation at e = 0,
⇒ dx/de (at e = 0) = -p(x(0)) / f'(x(0)) + 0
⇒ = - p(x*) / f'(x*)

(b)

(b) Given :- $f(x) = (x-1)(x-2)(x-3) --- (x-20)$, $p(x) = x^{19}$

To show :- $\left.\frac{dx}{d\varepsilon}\right|_{\varepsilon=0,\, x^*=j} = -\prod_{k\neq j} \frac{j}{j-k}$

⇒ So, $f'(x) = (x-2)(x-3) \_\_ (x-20) + (x-1)(x-3) \_ (x-20)$
$+ \_\_\_ + (x-1)(x-2)(x-3) \_\_ (x-19)$

⇒ Now, from given statement, $p(x) = x^{19} \Rightarrow p(x^*) = (x^*)^{19}$
$\Rightarrow p(j) = (j)^{19}$

⇒ Now, from the result of part (a) we have,

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

⇒ So we have to take $x^* = j$,

∴ $f'(j) = (j-2)(j-3) -- (j-20) + \_\_ + (j-1) \_ (j-19)$

⇒ $f'(j) = \prod_{k\neq j} (j-k)$ $\quad (1 \leq k \leq 20)$

⇒ This above eqn we build by on analysis i.e., if suppose $j=3$ then we have $f'(3)$,

⇒ $f'(3) = (j-1)(j-2)(j-4) \_ \_ (j-20)$

⇒ because only one term will not contain $(j-3)$ and all other contains $(j-3)$ and this will go for any value let's say k where k belongs to range $[1, 20]$

∴ $$\left.\frac{dx}{d\varepsilon}\right|_{\varepsilon=0,\, x^*=j} = \frac{-(j)^{19}}{\prod_{k\neq j} j-k}$$

⇒ Since there are total 19 terms in product,

∴ $$\left.\frac{dx}{d\varepsilon}\right|_{\varepsilon=0,\, x^*=j} = -\prod_{k\neq j} \frac{j}{j-k}$$ [Hence Proved]

(c)

(c) for $x^* = 1$,

$$\Rightarrow \left.\frac{du}{d\varepsilon}\right|_{\varepsilon=0,\,x^*=1} = -\prod_{k\neq 1} \frac{1}{1-k}$$

$$= -\left(\frac{1}{(1-2)(1-3)\,\cdots\,(1-20)}\right)$$

⇒ So there are 19 terms in denominator, so overall will be negative sign out,

$$\Rightarrow \left.\frac{du}{d\varepsilon}\right|_{\varepsilon=0,\,x^*=1} = +\frac{1}{(1\times 2\times 3 - 19)} = \frac{1}{1\times 2\times 3\times\cdots\times 19}$$

for $x^* = 20$,

$$\left.\frac{du}{d\varepsilon}\right|_{\varepsilon=0,\,x^*=20} = -\prod_{k\neq 20} \frac{20}{(20-k)}$$

$$\left.\frac{du}{d\varepsilon}\right|_{\varepsilon=0,\,x^*=20} = \frac{-(20)^{19}}{1\times 2\times 3\,\cdots\times 19}$$

⇒ So, denominator of both the terms are same so magnitude wise, value for $x^* = 20$ we get is greater than value of $x^* = 1$. Hence, $x^* = 1$ will be stable.

Ans - 4

(a) Observation and Explanation :-

We know that in IEEE double precision representation we have the Underflow here as `1.1125369292536007e-308` with a precision of 16 values after decimal and also IEEE allows denormal floating point numbers which added 16 more exponential values to 308 so we have a total 324. Hence, we have a bigger range of representable value and hence our second last value we see is `5e-324`, after that the number becomes `0.0` because we can't represent any number smaller than `5e-324` because of limiting exponent bits we have.

(b) Observation and Explanation :-

We know that in IEEE double precision has an epsilon machine of 2.22e-16 in binary64.So, if we add a number less than half of the epsilon machine to 1 it will remain equal to 1 (Rounding off). In our case we reach eps value at `1.1102230246251565e-16` which is less than half of an epsilon machine thus adding it to the number will not make a difference because of rounding off and thus our program ends by reporting eps value.

(c) Observation and Explanation :-

We know that in IEEE double precision representation we have the Overflow as 1.7976931348623157e+308.So any number above it will be infinite for our machine which is represented as “inf”. In our program we have the second last value as `8.98846567431158e+307` and the next value according to program will be 8.98846567431158e+307 * 2 which is greater than our Overflow value and hence it will be represented by “inf”.

Ans-5

```
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (7.0685834705770345, 0.9999999999999994, 0.0000000000000006, 0.0000000000000003, 14.1371669411540690)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (63.6172512351933079, 0.9999999999999897, 0.0000000000000103, 0.0000000000000052, 127.2345024703866301)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (629.1039288813560688, 0.9999999999999456, 0.0000000000000544, 0.0000000000000272, 1258.2078577627119103)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283.9707053429829102, 0.9999999999979700, 0.0000000000020300, 0.0000000000010150, 12567.9414106859658204)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62832.6384699592599645, 0.9999999999954939, 0.0000000000045061, 0.0000000000022530, 125665.2769399185199291)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628319.3161161219468340, 0.9999999998033864, 0.0000000001966136, 0.0000000000983068, 1256638.6322322436608374)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283186.0925777498632669, 0.9999999999777867, 0.0000000000222133, 0.0000000000111067, 12566372.1851555015891790)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62831853.8571940287947655, 1.0000000012561285, 0.0000000012561285, 0.0000000006280643, 125663707.7143880575895309)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628318531.5033566951751709, 0.9999997681704147, 0.0000002318295853, 0.0000001159148195, 1256637063.0067474842071533)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283185307.9649848937988281, 1.0000005069523146, 0.0000005069523146, 0.0000002534760288, 12566370615.9315834045410156)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62831853072.5812606811523438, 0.9999954970141940, 0.0000045029858060, 0.0000022515030415, 125663706146.4365539550781250)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628318530718.7440185546875000, 0.9999453990134111, 0.0000546009865889, 0.0000273019839690, 1256637063310.7758789062500000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283185307180.3710937500000000, 0.9984385423410914, 0.0015614576589086, 0.0007819488562304, 12566385957667.0527343750000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62831853071796.6484375000000000, 0.9965461389148867, 0.0034538610851133, 0.0017329054211049, 125664458272343.4218750000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628318530717959.3750000000000000, 0.8900802593986616, 0.1099197406013384, 0.0613308288477113, 1265166139489325.7500000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283185307179587.0000000000000000, 0.5766517306609554, 0.4233482693390446, 0.3177032810059763, 14519188958191300.0000000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62831853071795864.0000000000000000, -0.9682197486526081, 1.9682197486526081, 1.0158818422171154, 125729248279028768.0000000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628318530717958656.0000000000000000, -2.0518446488895292, 3.0518446488895292, 0.5857592547924637, 1595433312584221184.0000000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (6283185307179586560.0000000000000000, 5.5747653335805616, 4.5747653335805616, 0.1426137576491920, 36154359831181316096.0000000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (62831853071795863552.0000000000000000, -8.7118803024535953, 9.7118803024535953, 0.1262974912628313, 554595786581180809216.0000000000000000)
(x, tan(x), Abs. Forward Error, Abs. Backward Error, Relative Condition Number) = (628318530717958668288.0000000000000000, -1.0506941034214516, 2.0506941034214519, 0.9746839085961986, 1258173865379857956864.0000000000000000)
```

[Note :- In above output we show the (x,tan(x),Absolute Forward Error,Absolute Backward Error,Relative Condition Number)]

So, we have x = π/4+2π ×10j, j = 0, 1, 2, ... , 20

When we are calculating value of x for each iteration there is some perturbations added because of pie which we rounding off till 16 decimal places.So as our input x perturbed then our tan(x) also changes from expected result and exactly what happens with our computation also and also by computing relative condition number, we can analyze that their will be bigger changes happen if we have small changes in input x at each iteration because relative condition number keeps going on increasing as iteration count increasing which is value of j.

Ans - 6

(c)

Date / /

A- 6(c) Given :- $E = uv^T$

» To show :- $\|E\|_F = \|E\|_2 = \|u\|_2 \|v\|_2$

» We have given that $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$

» So, $E \in \mathbb{R}^{m \times n}$

» from Frobenius norm :-

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

$$= \sqrt{\operatorname{Tr} E^T E} \quad \text{(by definition)}$$

$$= \sqrt{\operatorname{Tr} (uv^T)^T(uv^T)}$$

$$= \sqrt{\operatorname{Tr} (u^Tv)(uv^T)}$$

$$= \sqrt{\operatorname{Tr} \underbrace{(u^Tu)}_{\text{This will be } 1\times1 \text{ matrix}} . \underbrace{(v^Tv)}_{\text{This also } 1\times1 \text{ matrix}}}$$

$$= \sqrt{\operatorname{Tr}(u^Tu)} \times \sqrt{\operatorname{Tr}(v^Tv)}$$

$$= \|u\|_2 . \|v\|_2 \quad \text{[Hence proved]}$$

Ans - 7

Output from 1st Equation :-

```
0.5822073316515288
0.5797135815734098
0.5788814056433012
0.5784651440685238
0.5782153315683285
0.5780487667534508
0.5779297805478292
0.5778405346932214
0.577771117576444
0.5777155815682065
0.5776701414855578
0.5776322736978301
0.5776002309764809
0.5775727652416682
0.5775489611978291
0.5775281323494514
0.5775097537135299
0.5774934169591495
0.5774787997122512
0.5774656440682016
0.577453741243179
0.5774429204111771
0.5774330404528918
0.5774239837672805
0.5774156515681996
0.5774079602664202
0.5774008386555254
0.5773942257008384
0.5773880687857824
0.5773823223089227
0.5773769465525795
0.5773719067634975
0.5773671724007521
0.5773627165162711
0.5773585152416505
0.5773545473603523
0.5773507939494777
0.5773472380778735
0.5773438645508655
0.5773406596931707
0.5773376111636459
0.5773347077964406
0.577331939464333
0.5773292969607322
0.5773267718973916
0.5773243566154296
0.5773220441077846
0.5773198279512748
0.5773177022470506
0.577315661568166
Final Result : 0.577315661568166
```

Output from 2nd equation :-

```
0.5772197901404903
0.5772167013748222
0.5772161263242399
0.5772159246680912
0.5772158312352449
0.577215780449559
0.5772157498141715
0.5772157299243794
0.5772157162847442
0.5772157065265553
0.5772156993055031
0.5772156938126143
0.57721568953740
0.5772156861448545
0.5772156834077089
0.5772156811674058
0.5772156793105871
0.5772156777544755
0.5772156764374801
0.5772156753129938
0.5772156743452568
0.577215673506438
0.5772156727746101
0.5772156721323229
0.5772156715655328
0.5772156710628664
0.5772156706149998
0.5772156702142466
0.5772156698542288
0.5772156695296005
0.5772156692358834
0.5772156689692576
0.5772156687264989
0.5772156685048309
0.5772156683019034
0.5772156681156311
0.57721566794427
0.5772156677862554
0.5772156676402354
0.5772156675050191
0.5772156673795781
0.5772156672629993
0.5772156671544533
0.5772156670532187
0.5772156669586632
0.5772156668702006
0.5772156667873283
0.5772156667095789
0.5772156666365351
0.5772156665678327
Final Result : 0.5772156665678327
```

Ans - 8

1. Operations on Matrix Generated randomly :-

Condition Number for all Matrix :-

```
Random Matrix Generated with size = 10 condtion number = 98.0532383479406
Random Matrix Generated with size = 20 condtion number = 1289.0327615231015
Random Matrix Generated with size = 30 condtion number = 269.9696210539937
Random Matrix Generated with size = 40 condtion number = 545.1388464888645
```

Error from un-pivoted Gaussian solve:-

```
Random Matrix Generated with size = 10 error from un-pivoted = 2.0011750770887195e-15
Random Matrix Generated with size = 20 error from un-pivoted = 3.953932145876932e-11
Random Matrix Generated with size = 30 error from un-pivoted = 1.791850892975832e-14
Random Matrix Generated with size = 40 error from un-pivoted = 8.004700308354878e-15
```

Residual from un-pivoted Gaussian solve:-

```
Random Matrix Generated with size = 10 residual from un-pivoted = 4.178589897868444e-16
Random Matrix Generated with size = 20 residual from un-pivoted = 8.790058916105379e-12
Random Matrix Generated with size = 30 residual from un-pivoted = 4.036937682128868e-15
Random Matrix Generated with size = 40 residual from un-pivoted = 1.2054961286175626e-15
```

Error from partial-pivot Gaussain solve:-

```
Random Matrix Generated with size = 10 error from pivoted = 8.426000324584082e-16
Random Matrix Generated with size = 20 error from pivoted = 7.3731264351444e-15
Random Matrix Generated with size = 30 error from pivoted = 8.918714851915906e-16
Random Matrix Generated with size = 40 error from pivoted = 3.510833468576701e-17
```

Residual from partial-pivot Gaussain solve:-

```
Random Matrix Generated with size = 10 residual from pivoted = 1.8798126492503614e-16
Random Matrix Generated with size = 20 residual from pivoted = 1.6523367074666114e-15
Random Matrix Generated with size = 30 residual from pivoted = 2.0589258583777352e-16
Random Matrix Generated with size = 40 residual from pivoted = 3.369464112752124e-17
```

Error from np.linalg.solve :-

```
Random Matrix Generated with size = 10 error from np.linalg.solve = 6.283185307179587e+20
Random Matrix Generated with size = 20 error from np.linalg.solve = 6.283185307179585e+20
Random Matrix Generated with size = 30 error from np.linalg.solve = 6.283185307179585e+20
Random Matrix Generated with size = 40 error from np.linalg.solve = 6.283185307179587e+20
```

Residual from np.linalg.solve :-

```
Random Matrix Generated with size = 10 residual from np.linalg.solve = 7.896792643347635e-17
Random Matrix Generated with size = 20 residual from np.linalg.solve = 1.1719854547815177e-16
Random Matrix Generated with size = 30 residual from np.linalg.solve = 1.1519749576485322e-16
Random Matrix Generated with size = 40 residual from np.linalg.solve = 2.1042296639812252e-16
```

2. Operations on Hilbert Matrix Generated :-
Condition Number for all Matrix :-

```
Hilbert Matrix Generated with size = 10 condtion number = 16024416992541.715
Hilbert Matrix Generated with size = 20 condtion number = 1.3553657908688225e+18
Hilbert Matrix Generated with size = 30 condtion number = 5.507991645999902e+18
Hilbert Matrix Generated with size = 40 condtion number = 6.507249058549335e+18
```

Error from un-pivoted Gaussian solve:-

```
Hilbert Matrix Generated with size = 10 error from un-pivoted = 3.111654735330855e-06
Hilbert Matrix Generated with size = 20 error from un-pivoted = 3.1677370090950725
Hilbert Matrix Generated with size = 30 error from un-pivoted = 55.32345776870196
Hilbert Matrix Generated with size = 40 error from un-pivoted = 48.38290871975059
```

Residual from un-pivoted Gaussian solve:-

```
Hilbert Matrix Generated with size = 10 residual from un-pivoted = 4.0457706053662927e-07
Hilbert Matrix Generated with size = 20 residual from un-pivoted = 0.00013221820585089083
Hilbert Matrix Generated with size = 30 residual from un-pivoted = 4.1552919376982135e-05
Hilbert Matrix Generated with size = 40 residual from un-pivoted = 4.904702357264281e-05
```

Error from partial-pivot Gaussain solve:-

```
Hilbert Matrix Generated with size = 10 error from pivoted = 8.537716491350523e-06
Hilbert Matrix Generated with size = 20 error from pivoted = 44.53592826916291
Hilbert Matrix Generated with size = 30 error from pivoted = 58.42684891829501
Hilbert Matrix Generated with size = 40 error from pivoted = 20.24756974862636
```

Residual from partial-pivot Gaussain solve:-

```
Hilbert Matrix Generated with size = 10 residual from pivoted = 1.110073107585775e-06
Hilbert Matrix Generated with size = 20 residual from pivoted = 0.000550676162559448
Hilbert Matrix Generated with size = 30 residual from pivoted = 0.00014317655841732306
Hilbert Matrix Generated with size = 40 residual from pivoted = 0.0001039159360533046
```

Error from np.linalg.solve :-

```
Hilbert Matrix Generated with size = 10 error from np.linalg.solve = 6.283185307179587e+20
Hilbert Matrix Generated with size = 20 error from np.linalg.solve = 6.283185307179585e+20
Hilbert Matrix Generated with size = 30 error from np.linalg.solve = 6.283185307179585e+20
Hilbert Matrix Generated with size = 40 error from np.linalg.solve = 6.283185307179587e+20
```

Residual from np.linalg.solve :-

```
Hilbert Matrix Generated with size = 10 residual from np.linalg.solve = 1.0026063529445472e-16
Hilbert Matrix Generated with size = 20 residual from np.linalg.solve = 3.257304536028333e-16
Hilbert Matrix Generated with size = 30 residual from np.linalg.solve = 2.6419110942075113e-16
Hilbert Matrix Generated with size = 40 residual from np.linalg.solve = 2.3921330329362507e-15
```

3. Operations on Matrix which contains only 1 and -1 :-
Condition Number for all Matrix :-

```
Matrix Generated with size = 10 condtion number which contains only 1 and -1 = 6.313751514675043
Matrix Generated with size = 20 condtion number which contains only 1 and -1 = 12.70620473617471
Matrix Generated with size = 30 condtion number which contains only 1 and -1 = 19.081136687728215
Matrix Generated with size = 40 condtion number which contains only 1 and -1 = 25.451699579357086
```

Error from un-pivoted Gaussian solve:-

```
Matrix Generated with size = 10 error from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 20 error from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 30 error from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 40 error from un-pivoted which contains only 1 and -1 = 0.0
```

Residual from un-pivoted Gaussian solve:-

```
Matrix Generated with size = 10 residual from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 20 residual from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 30 residual from un-pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 40 residual from un-pivoted which contains only 1 and -1 = 0.0
```

Error from partial-pivot Gaussain solve:-

```
Matrix Generated with size = 10 error from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 20 error from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 30 error from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 40 error from pivoted which contains only 1 and -1 = 0.0
```

Residual from partial-pivot Gaussain solve:-

```
Matrix Generated with size = 10 residual from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 20 residual from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 30 residual from pivoted which contains only 1 and -1 = 0.0
Matrix Generated with size = 40 residual from pivoted which contains only 1 and -1 = 0.0
```

Error from np.linalg.solve :-

```
Matrix Generated with size = 10 error from np.linalg.solve which contains only 1 and -1 = 6.283185307179587e+20
Matrix Generated with size = 20 error from np.linalg.solve which contains only 1 and -1 = 6.283185307179585e+20
Matrix Generated with size = 30 error from np.linalg.solve which contains only 1 and -1 = 6.283185307179585e+20
Matrix Generated with size = 40 error from np.linalg.solve which contains only 1 and -1 = 6.283185307179587e+20
```

Residual from np.linalg.solve :-

```
Matrix Generated with size = 10 residual from np.linalg.solve which contains only 1 and -1 = 0.0
Matrix Generated with size = 20 residual from np.linalg.solve which contains only 1 and -1 = 0.0
Matrix Generated with size = 30 residual from np.linalg.solve which contains only 1 and -1 = 0.0
Matrix Generated with size = 40 residual from np.linalg.solve which contains only 1 and -1 = 0.0
```