HW1

Graded

Student

Manshi Sagar

Total Points

45.5 / 72 pts

Question 1

Q1

Resolved **2** / 10 pts

+ 0 pts Incorrect

+ 7 pts Show that condition number is $\frac{|x|+|y|}{|y|}$ (assuming $|x| \geq |y|$).

+ 2 pts Say that the function is ill-conditioned when $|x| \gg |y|$ (or when $|y| \gg |x|$).

+ 1 pt Intuitive explanation when $|x| \gg |y|$.

✔ **+ 1 pt** Used different norms for condition number derivation for $||J_f(x,y)||$ and $||(x,y)||$. (exclusive with 2)

✔ **+ 1 pt** Say that the function is ill-conditioned when either x or y tends to zero. (exclusive with 3)

+ 0.5 pts Intuitive explanation when when either x or y tends to zero. (exclusive with 4)

$||J_f(x,y)||$ is a *matrix norm*, not a *vector norm* as you have written.

Regrade Request **Submitted on: Mar 06**

In the first page where i have calculated Kf(x, y)
i have got the expression
(|x| + |y|)(xΔy + yΔx) / |x| |y| (Δx+Δy)

in this expression if we assume x>>y, we will get the expression
(|x| + |y|)/ |x| |y|
as Δx~Δy,

i did not take the case x>>y, so i could not reach the final expression
please give some partial marks for previous steps

i have showed the cases when x, y are both close to 0

or when both x, y are large

i have also explained that when x , y tend to zero, Kf is very high

Reaching the final expression from your initial expression is what the marks were for, we can't give partial marks there.

Reviewed on: Mar 11

Question 2

Q2 Resolved **6** / 15 pts

+ **0 pts** Incorrect

+ **5 pts** Part (i)

✔ + **2.5 pts** Part (ii) - Not Backward Stable

✔ + **2.5 pts** Part (ii) - Is Stable

+ **4 pts** Part (iii) - Not Backward Stable

+ **1 pt** Part (iii) - Stable

+ **1 pt** Part (i) : You can't use approximately equal when the entire point of the question was to show this relationship. +1 mark

Part (iii) backward stability : How does high degree matter?

stability : lots of words, no working. also, what is the xhat you're claiming will satisfy these assumptions?

Regrade Request **Submitted on: Mar 06**

Part (iii)
backward stability:
degree will matter as the degree of denominator is only 1
Say degree of numerstor is d

then degree of x in backward error expression is (d-1)
if x is large , x^(d-1) is very large, hence not backward stable

The answer still is incorrect. The sin inverse expansion only holds for $|x| < 1$. So how can x be very large? :-)

Reviewed on: Mar 11

Question 3

Q3 **4** / 10 pts

✔ + **2 pts** i) correct

+ **0 pts** (i) incorrect

+ **5 pts** (ii) correct

✔ + **0 pts** (ii) incorrect

+ **3 pts** (iii) correct

✔ + **0 pts** (iii) incorrect

+ **2 pts** (ii) +1.5, (iii) +0.5

Marks deducted for missing/incorrect calculations

Question 4

Q4

Resolved 8.5 / 10 pts

✔ + 2 pts Part (i) correct

+ 1 pt Part (i) errors in calculating Kf

+ 1 pt Part (i) Informal proof for ill-conditioning

+ 0 pts Part (i) incorrect

+ 3 pts Part (ii) correct

+ 2 pts Part (ii) minor errors in calculation

+ 1.5 pts Part (ii) calculations missing

+ 1.5 pts Part (ii) errors in calculation

✔ + 1.5 pts Part (ii) mentioned cancellation errors, without calculations

+ 0 pts Part (ii) incorrect

✔ + 5 pts Part (iii) correct

+ 3.5 pts Part (iii) Minor errors in calculations

+ 2.5 pts Part (iii) calculations missing

+ 1.5 pts Part (iii) errors in calculations

+ 0 pts Part (iii) incorrect

1 You have not shown $\frac{|y-f(\hat{x})|}{f(\hat{x})}$ to be close to machine precision. The claim that $f(\hat{x}) = y$ does not hold if you substitute the expressions

Regrade Request Submitted on: Mar 06

in part c, i am calculating backward error to check backward stability,
so i wrote an expression for x^ and calculated backward error.

i did not use forward error, so i did not calculate the expression (y-f(x^))/f(x^)

i think that to show backward stability, this is what we are supposed to do.
and backward stability implies stability, so i dont think we need to show that forward error is close to machine epsilon.

please correct me if i am wrong and tell me how to show stability and backward stability correctly

the value of y considers all the floating point errors (hence the epsilons)

Regraded

Reviewed on: Apr 29

Question 5

Q5

Resolved 13 / 15 pts

✔ + 15 pts Correct

+ 2.5 pts A) Single Precision

+ 2.5 pts A) Double Precision

+ 1 pt B) Fib plot Single Precision

+ 1 pt B) Fib Plot Double Precision

+ 2 pts B) explanation

+ 2 pts C) Pib Plot Single Precision

+ 2 pts C) Pib Plot Double Precision

+ 2 pts C) explanation

+ 0 pts Unattempted/Incorrect

− 1.5 pts Late

💬 − 2 pts Bad Scaling

Regrade Request **Submitted on: Mar 06**

Can you please explain why are marks deducted for bad scaling?

I suppose this is correct scaling.

you should have used a log scale. It is not apparent where values first become non-zero

Reviewed on: Mar 06

Regrade Request **Submitted on: Mar 11**

But I have used log scale only

No, all of your graphs are linear scale for b and c

Reviewed on: Mar 11

Question 6

Q6

12 / 12 pts

- ✔ **+ 2 pts** Code
- ✔ **+ 2 pts** Part a Observation
- ✔ **+ 3 pts** Part b Fixed Points
- ✔ **+ 3 pts** Part c Plots
- ✔ **+ 2 pts** Part c Observations

+ 0 pts Incorrect/Unattempted

− 1.2 pts Late

Question assigned to the following page: 1

Q1 $f(x,y) = x \cdot y$

$f(x+\Delta x, y+\Delta y) = (x+\Delta x)\cdot(y+\Delta y)$

$$K_f(x,y) = \lim_{\substack{\Delta x \to 0 \\ \Delta y \to 0}} \left| \frac{f(x+\Delta x, y+\Delta y) - f(x,y)}{f(x,y)} \right| \times \frac{|x|+|y|}{|\Delta x|+|\Delta y|}$$

$$= \left| \frac{(x+\Delta x)(y+\Delta y) - x\cdot y}{x\cdot y} \right| \times \frac{|x|+|y|}{|\Delta x|+|\Delta y|}$$

$$= \frac{|x\cdot\Delta y + \Delta x\cdot y + \Delta x\,\Delta y|}{|x\cdot y|} \times \frac{|x|+|y|}{|\Delta x|+|\Delta y|}$$

$$= \frac{|x|+|y|}{|x\cdot y|} \times \frac{|x\cdot\Delta y + y\cdot\Delta x + \cancel{\Delta x\cdot\Delta y}^{\,0}|}{|\Delta x|+|\Delta y|}$$

$$= \frac{|x|+|y|}{|x||y|} \times \frac{|\Delta y/y + \Delta x/x|}{\left|\frac{\Delta x}{x\cdot y}\right| + \left|\frac{\Delta y}{x\cdot y}\right|}$$

for $x \approx 0$ and $y \approx 0$

$x = 10^{-5}$ $y = 10^{-5}$ $\Delta x = 10^{-20}$ $\Delta y = 10^{-20}$

$$K_f(x,y) = \frac{10^{-5}+10^{-5}}{10^{-10}}\left(\frac{10^{-5}\cdot 10^{-20} + 10^{-5}\cdot 10^{-20} + 10^{-40}}{2\cdot 10^{-20}}\right)$$

$$\approx 10^5 \times 10^{-5} \approx 1$$

For $x = 10^{15}$ $y = 10^{15}$

$\Delta x = 10^{-20}$ $\Delta y = 10^{-20}$

$$K_f = \frac{2\times 10^{15}}{10^{30}} \times \frac{2\times 10^{-5} + 10^{-40}}{2\times 10^{-20}}$$

$$\approx 1$$

Question assigned to the following page: 1

Finding condition number using Jacobian

$$K = \frac{\|J\|}{\|f(x,y)\|} \cdot \|(x,y)\|$$

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} y & x \end{bmatrix}$$

$$\|J\| = |x| + |y|$$

$$K_f = \left| \frac{(|x|+|y|)}{x \cdot y} \times |x|+|y| \right|$$

$$= \left| \frac{|x|^2 + |y|^2 + 2|xy|}{|xy|} \right|$$

$$= \left| \frac{1}{|x|} + \frac{1}{|y|} + 2 \right|$$

We can see that for values of $|x| \simeq 0$ or $|y| \simeq 0$ K_f will be very high

We can also see intuitively that the product will tend to zero in such cases.
We can predict loss of accuracy in such cases because, to correctly represent the product, we will need 2 times the number of bits.
But half of these bits will have to be truncated

Question assigned to the following page: 2

Q2 (i) $K_f(x) = \dfrac{|f(\hat{x}) - f(x)|}{|f(x)|} \times \dfrac{|x|}{|x - \hat{x}|}$

Using the notion of stability, we know that

$$\frac{|x - \hat{x}|}{|x|} = O(\varepsilon_{macn})$$

and $$\frac{|y - f(\hat{x})|}{|f(\hat{x})|} = O(\varepsilon_{macn})$$

where y = value computed by algorithm

Relative error in output $= \dfrac{|f(x) - y|}{|f(x)|} \times \dfrac{K_f(x)}{K_f(x)}$

$$= K_f(x) \times \frac{|f(x) - y|}{|f(x)|} \times \frac{f(x)}{f(\hat{x}) - f(x)} \times \frac{|x - \hat{x}|}{|x|}$$

$$= K_f(x) \times \frac{|x - \hat{x}|}{|x|} \times \frac{|f(x) - y|}{|f(x) - f(\hat{x})|}$$

$\frac{|x - \hat{x}|}{|x|} \to O(\varepsilon_{macn})$; $\frac{|f(x) - y|}{|f(x) - f(\hat{x})|} \to 1$

According to stability, y is very close to $f(\hat{x})$

$$\Rightarrow y \approx f(\hat{x})$$

$$f(x) - y \approx f(x) - f(\hat{x})$$

$$\Rightarrow \frac{|f(x) - y|}{|f(x) - f(\hat{x})|} \approx 1$$

$\Rightarrow$ Relative error in output $\approx K_f(x) \times \varepsilon_{macn} \times 1$

$$= O\left(K_f(x) \cdot \varepsilon_{macn}\right)$$

Question assigned to the following page: 2

Q2 (ii) $f(x) = 1 + x$

BACKWARD STABILITY

$$y = fl(1 + fl(x))$$

$$= (1 + x(1+\varepsilon_1))(1+\varepsilon_2) \qquad \varepsilon_i = O(\varepsilon_{mach}),\ i = 1, 2$$

$$= (1+\varepsilon_2) + x(1+\varepsilon_1)(1+\varepsilon_2)$$

$$= 1 + (\varepsilon_2 + x(1+\varepsilon_1)(1+\varepsilon_2))$$

$$\hat{x} = \varepsilon_2 + x(1+\varepsilon_1)(1+\varepsilon_2)$$
$$= \varepsilon_2 + x(1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2)$$

$$\text{Backward Error} = \frac{|\hat{x} - x|}{|x|} = \frac{|\varepsilon_2 + x(\varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2)|}{|x|}$$

$$= \frac{|\varepsilon_2|}{|x|} + (\varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2)$$

Here, Backward error will be very large (close to infinity) for $|x|$ close to zero due to the term $|\varepsilon_2|/|x|$

Hence, the algorithm is not backward stable

STABILITY

$$\hat{x} = x(1+\varepsilon_1) \qquad \varepsilon_1 = O(\varepsilon_{mach})$$

$$f(\hat{x}) = 1 + x(1+\varepsilon_1)$$

$$y = 1 + \varepsilon_2 + x(1+\varepsilon_1)(1+\varepsilon_2) \qquad \varepsilon_2 = O(\varepsilon_{mach})$$

$$\left|\frac{y - f(\hat{x})}{f(\hat{x})}\right| = \left|\frac{\varepsilon_2 + x(1+\varepsilon_1)(\varepsilon_2)}{1 + x(1+\varepsilon_1)}\right|$$

for $x \approx 0$ $\quad \left|\frac{y - f(\hat{x})}{f(\hat{x})}\right| \approx O(\varepsilon_2) = O(\varepsilon_{mach})$

and for very large values of x, $\left|\frac{y - f(\hat{x})}{f(\hat{x})}\right| \approx \left|\frac{x(1+\varepsilon_1)(1+\varepsilon_2)}{x(1+\varepsilon_1)}\right|$

Question assigned to the following page: 2

$\simeq O(\varepsilon_{mach})$

Hence the algorithm is stable.

We can also see intuitively that the value computed by the algorithm will be very close to $1+x$ as the will not be much cancellation error

for values of x close to (-1)
we will see large error due to cancellation error

Question assigned to the following page: 2

Q2 (iii) $f(x) = \sin x$ $\quad x \in [0, 2\pi]$

BACKWARD STABILITY

$y = (1+\varepsilon_2)\sin(x(1+\varepsilon_1))$ $\quad |\varepsilon_1| \le \varepsilon_{mach}$, $|\varepsilon_2| \le \varepsilon_{mach}$

$= $ output of algorithm

$\hat{x} = \sin^{-1}(1+\varepsilon_2)\sin(x(1+\varepsilon_1))$

$$\frac{|\hat{x} - x|}{|x|} = \frac{\sin^{-1}(1+\varepsilon_2)\sin(x(1+\varepsilon_1)) - x}{x}$$

using the expansion of

$$\sin^{-1} a = a + \frac{1 \cdot a^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot a^5}{2 \cdot 4 \cdot 5} + \cdots$$

$$\sin a = a - \frac{a^3}{3!} + \frac{a^5}{5!} + \cdots$$

we can see that the numerator will have very high degree (25 if we take only 3 terms) but denominator has degree = 1.

Hence the backward error will be very high
$\Rightarrow$ Algorithm will be Backward unstable.

STABILITY

If we are given $y =$ value computed by our algorithm

then we can find $\hat{x}$ close to x which satisfies $\left| \frac{y - f(\hat{x})}{f(\hat{x})} \right| = O(\varepsilon_{mach})$ — (1)

using the polynomial expansions since sin x is a smooth function with a small range $[-1, 1]$,

if $\hat{x}$ satisfies $\frac{|\hat{x} - x|}{|x|} = O(\varepsilon_{mach})$, (1) will also hold

Question assigned to the following page: 3

Q3 $f(x) = \dfrac{e^x - 1}{x}$

(a) $K_f(x) = \dfrac{x \times x}{e^x - 1} \times \left((e^x - 1)\left(\dfrac{-1}{x^2}\right) + \dfrac{1}{x}(e^x) \right)$

$= \dfrac{x^2}{e^x - 1} \times \left(\dfrac{-e^x + 1}{x^2} + \dfrac{x \cdot e^x}{x^2} \right)$

$= \dfrac{1}{e^x - 1} \left(x \cdot e^x - e^x + 1 \right) = \dfrac{1}{e^x - 1} \left(x \cdot e^x - (e^x - 1) \right)$

$= \dfrac{x \cdot e^x}{e^x - 1} - 1$

Using Taylor series expansion

$e^x = 1 + x + \dfrac{x^2}{2!} + \dfrac{x^3}{3!} + \cdots$

$K_f(x) = \dfrac{x + x^2 + x^3/2! + x^4/3!}{x + \dfrac{x^2}{2!} + \dfrac{x^3}{3!}}$ using 4 terms

for values very close to zero say $x = 10^{-5}$ or $x = 10^{-10}$

$K_f(x)$ is almost 1.

Question assigned to the following page: 3

Q3 (b) $f(x) = \dfrac{e^x - 1}{x}$

$x \longrightarrow fl(x) = x(1+\varepsilon_1)$ $\quad \varepsilon_1 \leq \varepsilon_{mach}$

$e^x \longrightarrow fl(e^x) = \left(e^{x(1+\varepsilon_1)}\right)(1+\varepsilon_2)$ $\quad \varepsilon_2 \leq \varepsilon_{mach}$

$$y = \left(\frac{\left(e^{x(1+\varepsilon_1)}\right)(1+\varepsilon_2) - 1}{x(1+\varepsilon_1)}\right)(1+\varepsilon_3)(1+\varepsilon_4)$$

, let us consider $\hat{x} = 1+\varepsilon_1$

$$\text{Relative error} = \frac{|y - f(\hat{x})|}{|f(\hat{x})|}$$

$$= \frac{\left|\dfrac{(1+\varepsilon_4)(1+\varepsilon_3)(1+\varepsilon_2)\,e^{x(1+\varepsilon_1)} - (1+\varepsilon_3)(1+\varepsilon_4)}{x(1+\varepsilon_1)} - f(\hat{x})\right|}{|f(\hat{x})|}$$

$$= \left|\frac{e^{x(1+\varepsilon_1)}(\varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \cdots) - (\varepsilon_3 + \varepsilon_4 + \varepsilon_3\varepsilon_4)}{e^{x(1+\varepsilon_1)} - 1}\right|$$

for $x \simeq 0$ both numerator and denominator will tend to 0 leading to a infinite relative error

since the problem is well conditioned around 0, it must be unstable to result in infinite relative error

Question assigned to the following page: 3

Q3 (c) $f(x) = \dfrac{e^x - 1}{\ln e^x}$

$x \longrightarrow x(1+\varepsilon_1)$ $\quad \varepsilon_1 \leq \varepsilon_{mach}$

$e^x \longrightarrow (1+\varepsilon_2)\, e^{x(1+\varepsilon_1)}$ $\quad \varepsilon_2 \leq \varepsilon_{mach}$

$\ln e^x \longrightarrow (1+\varepsilon_3)(1+\varepsilon_2) \ln e^{x(1+\varepsilon_1)}$ $\quad \varepsilon_3 \leq \varepsilon_{mach}$

$$y \longrightarrow \frac{(1+\varepsilon_2)\, e^{x(1+\varepsilon_1)} - 1}{(1+\varepsilon_3)(1+\varepsilon_2) \ln e^{x(1+\varepsilon_1)}}$$

we can find $\hat{x}$, s.t. $f(\hat{x}) = y$

For Backward stability

$$\frac{|x - \hat{x}|}{|x|} = O(\varepsilon_{mach})$$

Question assigned to the following page: 4

Q4

(i) $f(x) = \frac{1}{1-x} + \frac{1}{1+x}$

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

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

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

$$\approx 1$$

The function is well conditioned.

Question assigned to the following page: 4

94 (b) $f(x) = \frac{1}{1-x} - \frac{1}{1+x}$

For values close to 0, condition no. = 1
This means that problem is well conditioned

But the backward around $x \simeq 0$ is very high because $\left(\frac{1}{1-x}\right)$ and $\left(\frac{1}{1+x}\right)$ have very close magnitudes and due to cancellation error, this will result in unstability.

$K_f(x) = \frac{\text{forward error}}{\text{backward error}}$

Since both numerator and denominator are very large we see that $K_f(x) \simeq 1$ but this does not ensure stability

Question assigned to the following page: 4

Q4 (c) $f(x) = \frac{2x}{1-x^2}$

Here, there will not be cancellation error due to subtraction of close magnitudes (which was the cause of instability around $x = 0$)

$x \rightarrow \quad fl(x) = x(1+\varepsilon_1)$

$2x \rightarrow fl(2 \cdot fl(x)) = 2x(1+\varepsilon_1)(1+\varepsilon_2)$

$x^2 \rightarrow fl(fl(x) \cdot fl(x)) = x^2(1+\varepsilon_1)^2(1+\varepsilon_3)$

$1 - x^2 \rightarrow fl(1 - fl(x^2)) = \left(1 - x^2(1+\varepsilon_1)^2(1+\varepsilon_3)\right)(1+\varepsilon_4)$

$$y = \frac{2x(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_5)}{\left(1-x^2(1+\varepsilon_1)^2(1+\varepsilon_3)\right)(1+\varepsilon_4)}$$

we can find $\hat{x}$ st

$$\frac{2\hat{x}}{1-\hat{x}^2} = y$$

the value will come out to be close to

$$\hat{x} = x(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_5)$$

$$\frac{|\hat{x}-x|}{|x|} = \frac{|x(\varepsilon_1 + \varepsilon_2 + \varepsilon_5 + \varepsilon_1\varepsilon_2 + \varepsilon_2\varepsilon_5 + \varepsilon_1\varepsilon_2\varepsilon_5)|}{|x|}$$

$$= O(\varepsilon_{mach})$$

Backward error $= O(\varepsilon_{mach})$ Hence, this algorithm is backward stable for all x

Question assigned to the following page: 5

COL726-Assignment 1

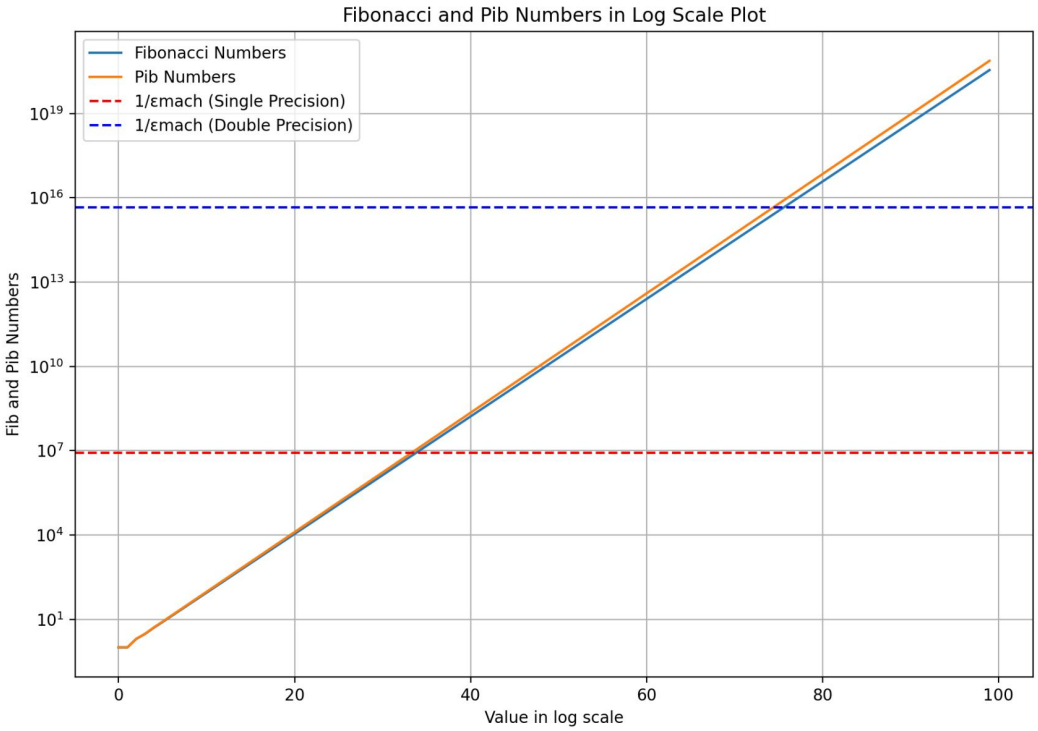
Manshi Sagar-2020CS50429

January 2024

1 Q5

1.1 Part a

SINGLE PRECISION



Both Fibonacci and the perturbated Fibonacci (pib) numbers grow exponentially, which is why, their graph is linear on a log scale plot.
For smaller values of n, the values of Fibonacci and Pib numbers are very close. But as n increases, due to the additional factor of c = 1/sqrt(3), Pib numbers become clearly larger than Fibonacci sequence.
In the log scale the difference looks small but in linear scale the difference would be very high. This is because the loss of precision in the c factor grows exponentially, so the dufference keeps on increasing as n increases.
The $1/\epsilon$ for single precision hits the sequences around 32-34.
$1/\epsilon$ for single precision = $2\hat{}$(24-1) = 8388608.0
fib(32) = 3524578.0
fib(33) = 5702887.0
fib(34) = 9227465.0
pib(32) = 4485154.253471081
pib(33) = 7313465.462491362
pib(34) = 11925292.65401772
We may see noticable changes in single precision around these fibonacci and pib numbers.

Question assigned to the following page: 5

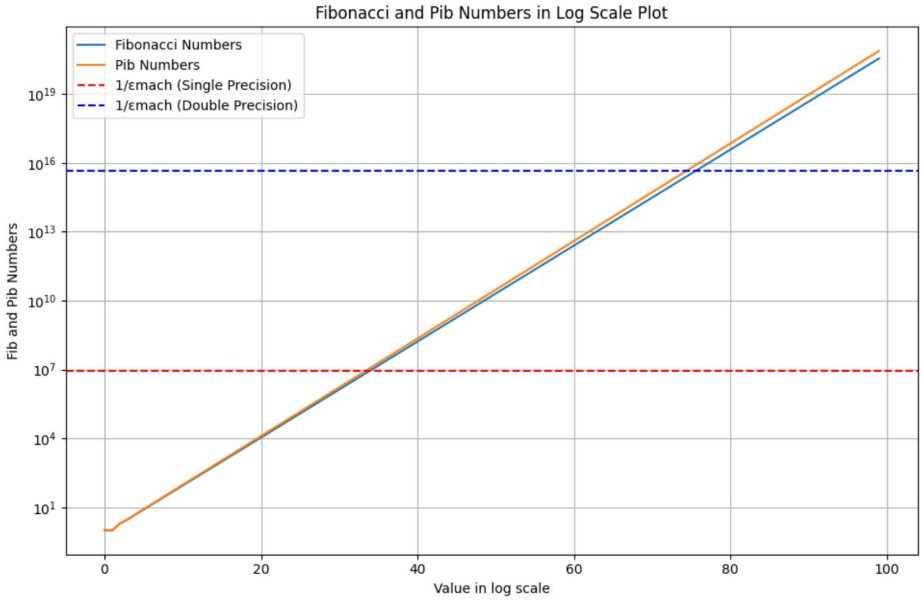
```
c = 1 + np.float32(np.sqrt(3)/100)

def generate_fibonacci(n):
    fib_numbers = [np.float32(1.0), np.float32(1.0)]
    while len(fib_numbers) < n:
        fib_numbers.append(fib_numbers[-1] + fib_numbers[-2])
    return fib_numbers


def generate_pib(n):
    pib_numbers = [np.float32(1.0), np.float32(1.0)]
    while len(pib_numbers) < n:
        pib_numbers.append(c*pib_numbers[-1] + pib_numbers[-2])
    return pib_numbers
```

DOUBLE PRECISION

$1/\epsilon$ for double precision = 4503599627370496.0
Due to more precision, the error due to the factor of c is less in double precision.



The $1/\epsilon$ for double precision hits the sequences around 74-76.
pib(74) = 3717939177735283.5
fib(75) = 3416454622906707.0
fib(76) = 5527939700884757.0
fib(74) = 2111485077978050.0
pib(75) = 6062449189171254.0
pib(76) = 9885393067046032.0
We may see noticable changes in duoble precision around these fibonacci and pib numbers.

```
c = 1 + np.float64(np.sqrt(3)/100)

def generate_fibonacci(n):
    fib_numbers = [np.float64(1.0), np.float64(1.0)]
    while len(fib_numbers) < n:
        fib_numbers.append(fib_numbers[-1] + fib_numbers[-2])
    return fib_numbers


def generate_pib(n):
```

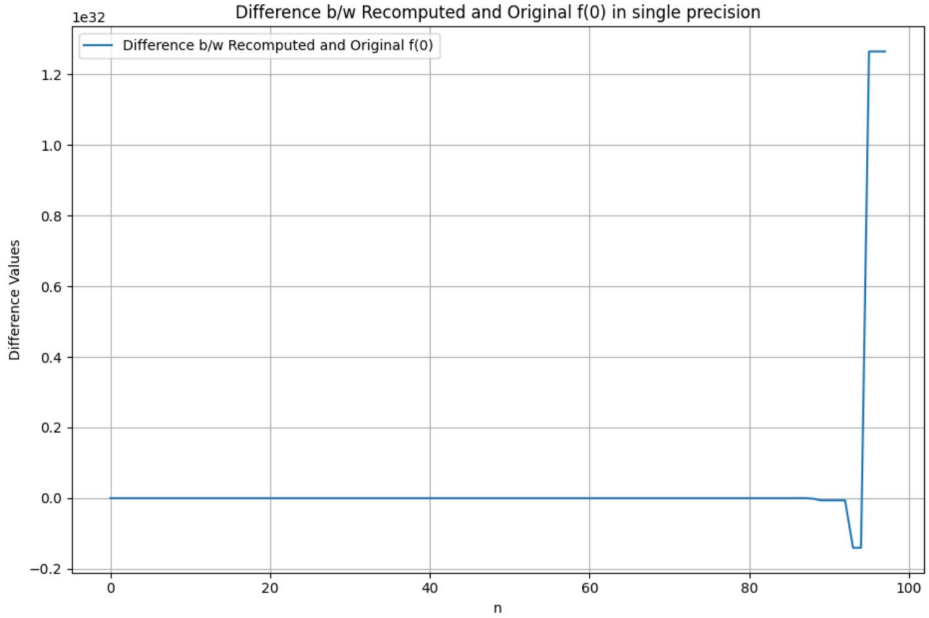
Question assigned to the following page: 5

```
pib_numbers = [np.float64(1.0), np.float64(1.0)]
while len(pib_numbers) < n:
    pib_numbers.append(c*pib_numbers[-1] + pib_numbers[-2])
return pib_numbers
```

1.2 Part b

```
def recompute_fibonacci(n):
refib_numbers = [fibonacci_numbers[n-1], fibonacci_numbers[n-2]]
while len(refib_numbers) < n:
    refib_numbers.append(refib_numbers[-2] - refib_numbers[-1])
return refib_numbers
```

SINGLE PRECISION

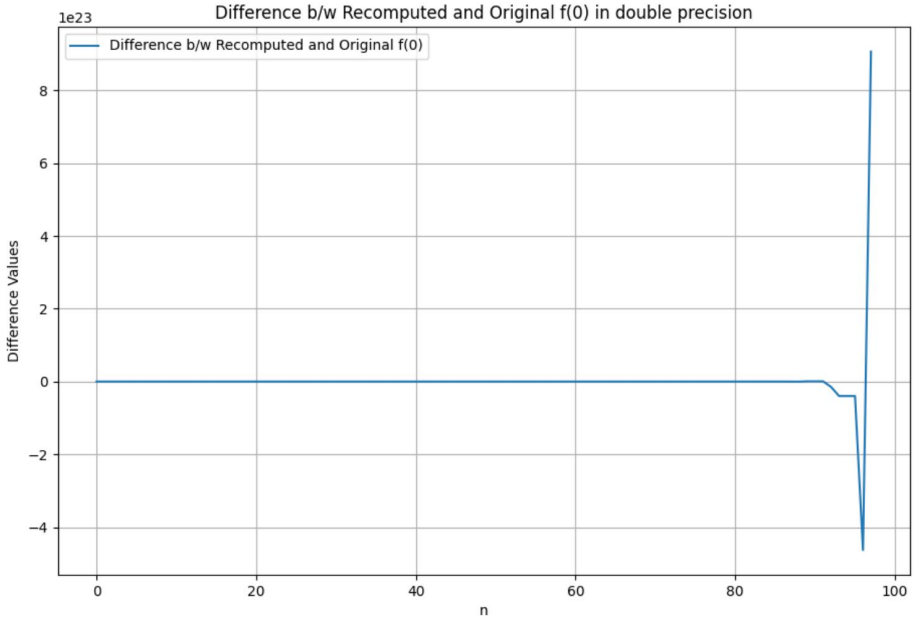


As we predicted in part a, the error in single precision starts to show at n=35.
For n=1 to 34, the difference between the recomputed f(0) and original f(0) is 0. Since only 24 bits can be stored in single precision but the value of fibonacci(35) exceeds 2^{24}, there is loss in precision due to **truncation error** and we see non-zero difference.
diff(33) = 0.0
diff(34) = 0.0
diff(35) = -9227465.0
diff(36) = -9227465.0
diff(37) = -9227465.0

We can see very large difference in the original and recomputed f[0] around n=90 because the numbers become very large and their values become indistinguishable due to limited 24-bit representation. Around n=90, the value fib(i) and fib(i+1) become very close. Thus, their subtraction results in very high **cancellation error**.

DOUBLE PRECISION

Question assigned to the following page: 5



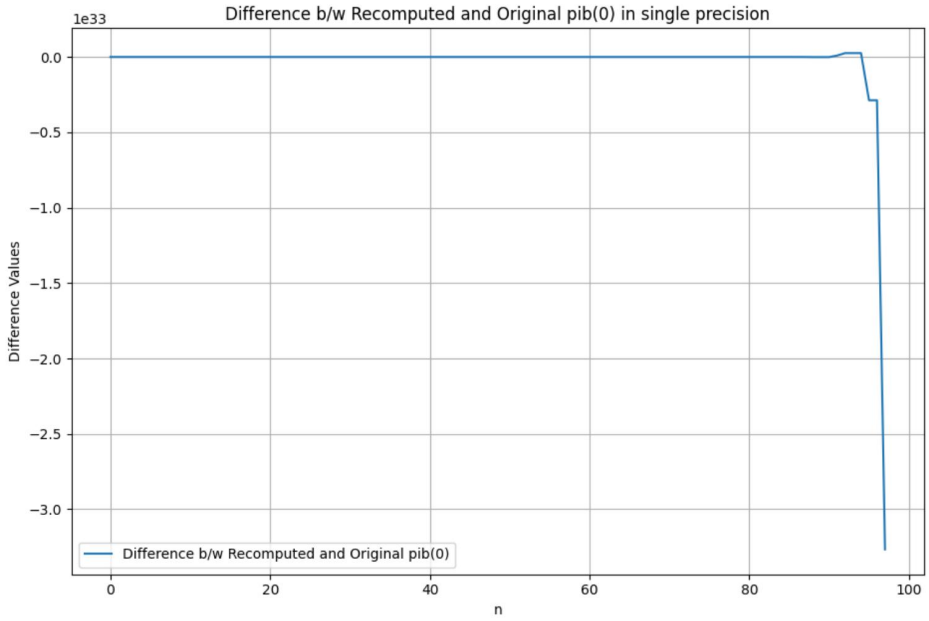
diff(75) = 0.0
diff(76) = 0.0
diff(77) = -5527939700884757.0
diff(78) = -5527939700884757.0
diff(79) = -5527939700884757.0
As we predicted in part a, the error in double precision starts to show at n=77.
For n=1 to 34, the difference between the recomputed f(0) and original f(0) is 0. Since only 48 bits can be stored in double precision but the value of fibonacci(77) exceeds 2^{48}, there is loss in precision due to **truncation error** and we see non-zero difference.
We can see very large negative difference in the original and recomputed f[0] around n=90 because the numbers become very large and go out of bounds.
We can see very large difference in the original and recomputed f[0] around n=90 because the numbers become very large and their values become indistinguishable due to limited 48-bit representation.
Around n=90, the value fib(i) and fib(i+1) become very close. Thus, their subtraction results in very high **cancellation error**.

1.3 Part c

```
def recompute_pib(n):
repib_numbers = [pib_numbers[n-1], pib_numbers[n-2]]
while len(repib_numbers) < n:
    repib_numbers.append(repib_numbers[-2] - c*repib_numbers[-1])
return repib_numbers
```

SINGLE PRECISION

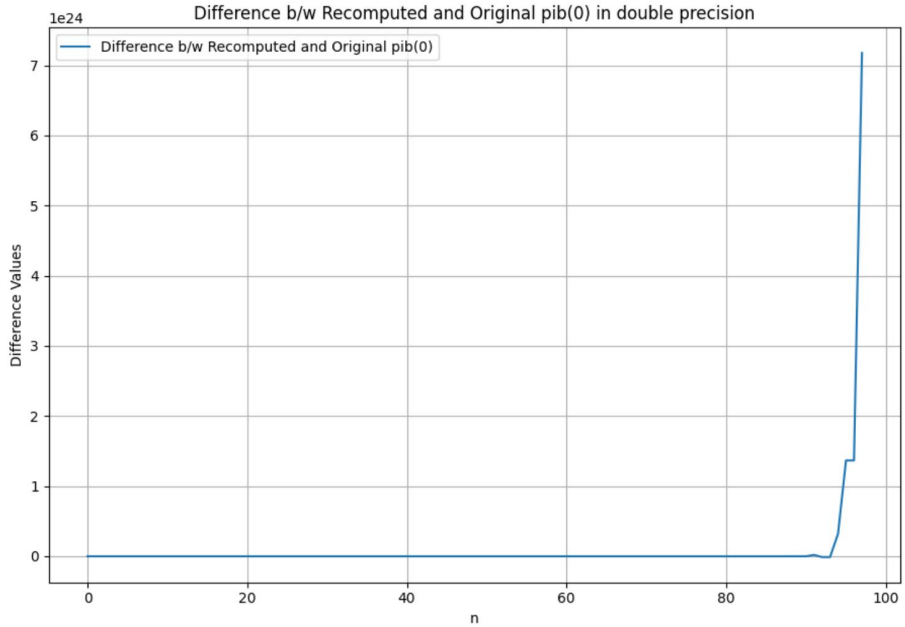
Question assigned to the following page: 5



diff(0) = 0.0
diff(1) = 1.1920928955078125e-07
diff(2) = 1.1920928955078125e-07

Here, we start to see the difference between the original and recomputed pib(0) at n=1 itself.
This is because we are using a factor of $sqrt(3)$.
$sqrt(3)$ is irrational and has a non-terminating decimal expansion. So, it can not be stored either in single precision(24 bits) or double precision(48 bits). So, there will always be some **truncation error**.

DOUBLE PRECISION



Questions assigned to the following page: 5 and 6

diff(3) = 0.0
diff(4) = 0.0
diff(5) = -4.440892098500626e-15
diff(6) = -4.440892098500626e-15
diff(7) = -5.240252676230739e-14
In double precision, since we can store more number of bits, the error is rather small. Here we see non-zero difference in the recomputed and original pib after n=5. This is because the difference in pib(1) to pib(4) must not be in the 48 bits actually stored in double precision. So, we see zero error for those values.
We see large positive difference in the recomputed and original pib after n=90.
Around n=90, the value pib(i) and pib(i+1) become very close. Thus, their subtraction results in very high **cancellation error**.

2 Q6

2.1 Part a

```
def generate_sequence(n):
    x_values = np.zeros(n, dtype=np.float32)
    x_values[0] = 11/2
    x_values[1] = 61/11

    for k in range(1, n-1):
        x_values[k+1] =  111- (1130-3000/x_values[k-1])/x_values[k]

    return x_values
```

I get the following values of xi:
x 1 = 5.5
x 2 = 5.5454545
x 3 = 5.590163
x 4 = 5.6334186
x 5 = 5.6744266
x 6 = 5.7094173
x 7 = 5.6806335
x 8 = 4.5765996
x 9 = -20.514648
x 10 = 134.1294
This inflation in error is due to huge cancellation error due to subtraction of numbers with very close absolute values.
The term $(1130 - 3000/y)/x$ keeps growing close to 111 and error keeps increasing. Finally, we see values very far from actual predicted value of 6. Observe x8, x9, x10, ...

2.2 Part b

The possible fixed points of F are: (5,5)
(6,6)
(100, 100)
We can verify this by using the equation: xk+1 = 111 (1130 3000/xk1)/xk
Let's substitute $x_k = 5$ and $x_{k-1} = 5$ into the recurrence relation

$$x_{k+1} = 111 - \frac{1130 - \frac{3000}{x_{k-1}}}{x_k}$$

and see what the result is:

$$x_{k+1} = 111 - \frac{1130 - \frac{3000}{5}}{5}$$

Simplifying the expression:

$$x_{k+1} = 111 - \frac{1130 - 600}{5}$$

Question assigned to the following page: 6

$$x_{k+1} = 111 - \frac{530}{5}$$

$$x_{k+1} = 111 - 106$$

$$x_{k+1} = 5$$

Therefore, if $x_k = 5$ and $x_{k-1} = 5$, the recurrence relation

$$x_{k+1} = 111 - \frac{1130 - \frac{3000}{x_{k-1}}}{x_k}$$

simplifies to $x_{k+1} = 5$. This means that if both x_k and x_{k-1} are equal to 5, the next iteration x_{k+1} remains 5.
if $x_k = 6$ and $x_{k-1} = 6$,

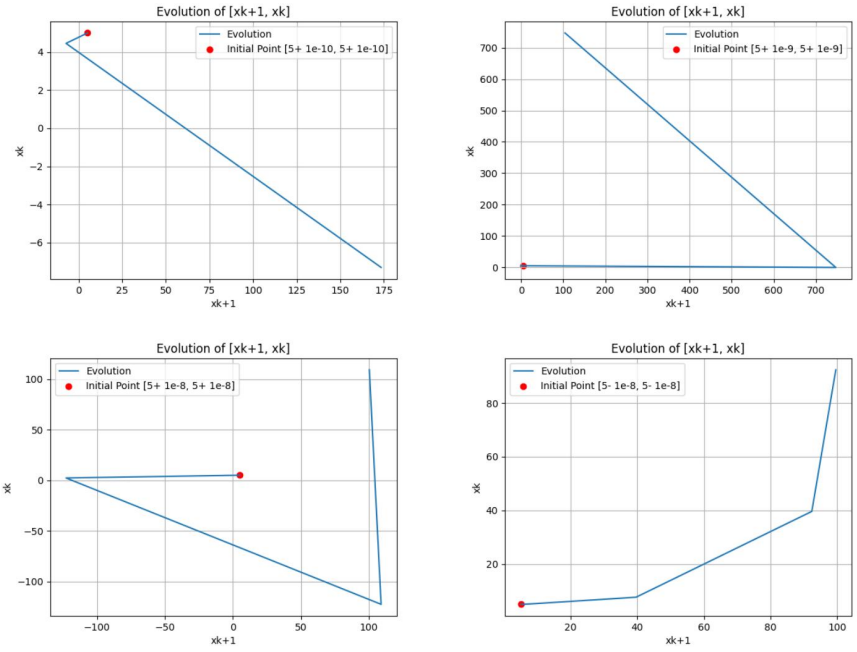
$$x_{k+1} = 111 - \frac{1130 - \frac{3000}{6}}{6} = 111 - \frac{1130 - 500}{6} = 111 - \frac{630}{6} = 111 - 105 = 6$$

Therefore, if $x_k = 6$ and $x_{k-1} = 6$, the recurrence relation $x_{k+1} = 111 - \frac{1130 - \frac{3000}{x_{k-1}}}{x_k}$ simplifies to $x_{k+1} = 6$.
if $x_k = 100$ and $x_{k-1} = 100$,

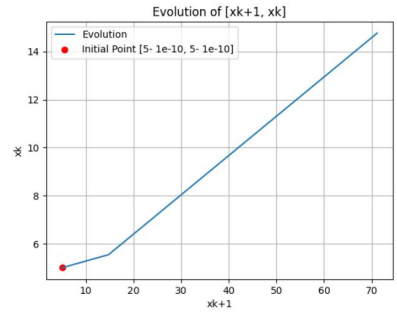
$$x_{k+1} = 111 - \frac{1130 - \frac{3000}{100}}{100} = 111 - \frac{1130 - 30}{100} = 111 - \frac{1100}{100} = 111 - 11 = 100$$

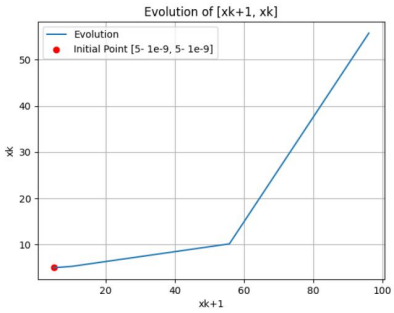
Therefore, if $x_k = 100$ and $x_{k-1} = 100$, the recurrence relation $x_{k+1} = 111 - \frac{1130 - \frac{3000}{x_{k-1}}}{x_k}$ simplifies to $x_{k+1} = 100$, maintaining a steady state.

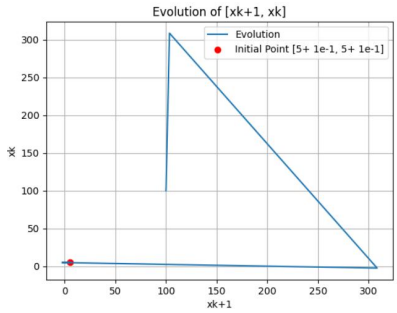
2.3 Part c

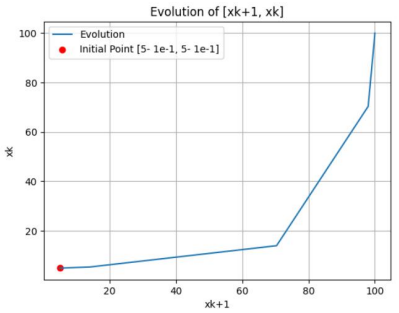


Question assigned to the following page: 6









We can see that the values of [xk+1, xk] are highly sensitive around x1=5, x2=5;
This is because of the high cancellation error due to the subtraction of values of very close magnitude. As values of x1 and x2 become slighly more than 5, the term $(1130 - 3000/xk - 1)/xk$ start to approach 111, this results in very large cancellation error. Hence, we see the large perturbation in the x-values.
As k increases, the values of xk, xk-1 start to approach 6 but due to the cancellation error also increasing, instead on monotonically increasing to 6, the values start to deviate. On increasing precision, the perturbation can be delayed for bigger values of k but the problem still remains.