COL726

Assignment -1

K.Laxman
2018CS50408

① $f: \mathbb{R}^2 \to \mathbb{R}$ by $f(x,y) = x - y$

Measure the size of input (x,y) by $|x| + |y|$

The relative condition number of a function measures how sensitive the output of function to small changes

here to find relative condition number using formula

$$K(x,y) = \|\nabla f(x,y)\| * \frac{\|x,y\|}{|f(x,y)|} \longrightarrow ③$$

$$\frac{\partial f}{\partial x} = 1 \longrightarrow ①$$

$$\frac{\partial f}{\partial y} = -1 \longrightarrow ②$$

from ① & ②

$$\nabla f = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

here vector norm of (x,y) is $|x| + |y|$ and norm of ∇f is 2

From eq ③

$$\text{relative condition number} = 2 * \frac{|x| + |y|}{|x - y|}$$

here Relative condition number is high when $|x-y|$ is small and $|x| + |y|$ is large

* It can be high when x and y are close in value but have larger magnitudes

In floating point numbers high relative condition number can occur when difference b/w input is less than machine epsilon (ε) leading sensitivity to small changes in input and unexpected results in graphs

$$* \frac{\text{ratio of absolute change in output}}{\text{ratio of small changes in input}}$$

② $f(x) = \frac{1}{1-x} - \frac{1}{1+x} \rightarrow ①$

ⓐ The computation will be unstable for values of x close to 1 (or) -1.

If $x \in (-\varepsilon, \varepsilon)$,
Assuming ε is 2^{-24} (single precision)
The value of $(1-x)$ and $(1+x)$ is nearly same. it will lead to truncation error and rounding error.
so, while taking $x \in (\varepsilon, \varepsilon)$ f is unstable.

ⓑ For the same above expression, ① can be written as

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

In this case the function $f(x)$ can be computed accurately in $x \in (-\varepsilon, \varepsilon)$

(3)

a) plot the numbers f_n, p_n on one log scale plot.

For single precision, ε_{mach} is 2^{-24} and for double precision $\varepsilon_{mach} = 2^{-53}$

$\frac{1}{\varepsilon_{mach}} = 2^{24}$ and $\frac{1}{\varepsilon_{mach}} = 2^{53}$ for double precision

(b) here to recompute f_K for $K = n-2, n-3, \ldots 0$.

$$f_{K+1} = f_K + f_{K-1}$$

$$\Rightarrow f_{K-1} = f_{K+1} - f_K \rightarrow (1)$$

~~and we compute~~

~~then~~. (1) is used to make a plot difference between original $f_0 = 1$ and recomputed f_0 as a function of n,

* The value of n ~~for~~ values results in low accuracy for recommend

the value of n will depend on precision of arithmetic used to compute the f_0 will depend on precision of arithmetic used to compute the recomputed f_0. generally large value of 'n' means less accurate recomputed f_0. and vice versa.

In case of single precision:- relative error will be larger than double precision, as the machine epsilon is larger

(c) Striking difference is that loss of precision for the fibonacci numbers is exponential as the ratio b/w consecutive fibonacci numbers increases rapidly. while loss of precision in case of pertubed Fib. number is much slower ~~case~~

④ consider the function $f(x) = 1-\cos x$

ⓐ relative condition number at $x = 0$

$$C_f(x) = \frac{x \cdot f'(x)}{f(x)} \quad \because C_f(x) \text{ denotes relative condition number}$$

$$\Rightarrow \frac{x \cdot \sin x}{1-\cos x}$$

$$\simeq \frac{x \cdot x}{x^2/2} \quad \text{when } x \to 0,$$

$$= 2$$

ⓑ The numerical evaluation of the formula $1-\cos(x)$ is highly unstable this is likely due to subtraction, of two very, similar values (1 and $\cos(x)$) which can result in a cancellation of significant digits.

ⓒ The stable algorithm for computing $f(x)$ is
The above $f(x) = 1-\cos x$ by differentiating $f(x)$

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

$$\Rightarrow 2\sin\left(\frac{x}{2}\right)^2$$

In this case the error is less/small between computed result and true result because in the case of $(1-\cos x)$ it has high error rate.

As, the error is small and does not grow excessively as

(5) For the case of polynomial in expanded form, it is more unstable and eratic. here we are calculating power operation once to each and every power (x), which gives different results for all points and adding, multiplication arises the Truncation error.

i.e. $x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1$

but, $(x-1)^6$ is more accurate & stable because we are calculating only one power compared to above.

(6) No, the graph doesn't confirm the expected behavior because initially we can see the graph is monotonically decreasing till $K=20$, but after that it is increasing as K increases bcz it is getting closer and closer to zero. After some time truncating error plays a role and gets accumulated which increases the function value in this case.

(7) There are some errors that may occur due to the program

i) Numerical errors:- as the integral is calculated recursively, floating point errors may occur and accumulate.

ii) Truncation error:- The program is only calculated the value of integral for $K=0$ to 20, so if we want more precision we need to increase the limit of K.

PLOTS FOR ASSIGNMENT1 COL726

K LAXMAN 2018CS50408

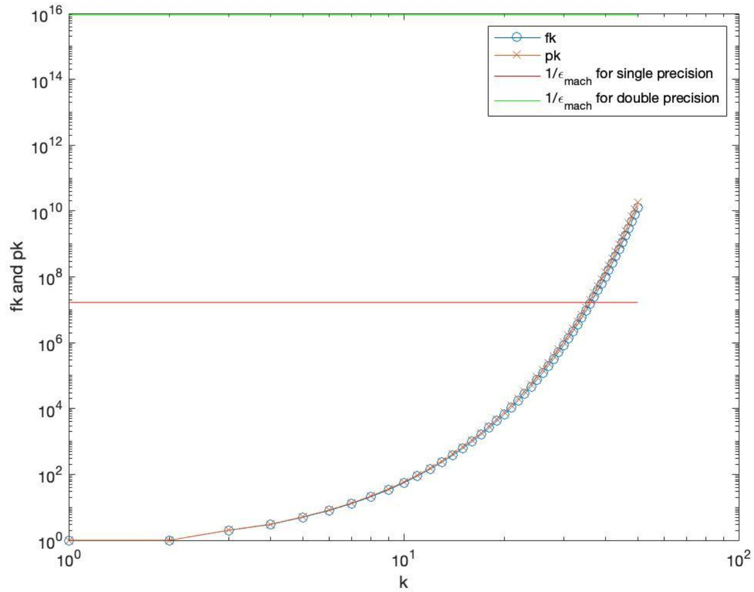


FIG:3(a)

https://drive.google.com/file/d/1orwzpdRux_k1qq2Qqo3H-2BHYDYWPOAI/view?usp=sharing
Google Drive link for MATLAB Code

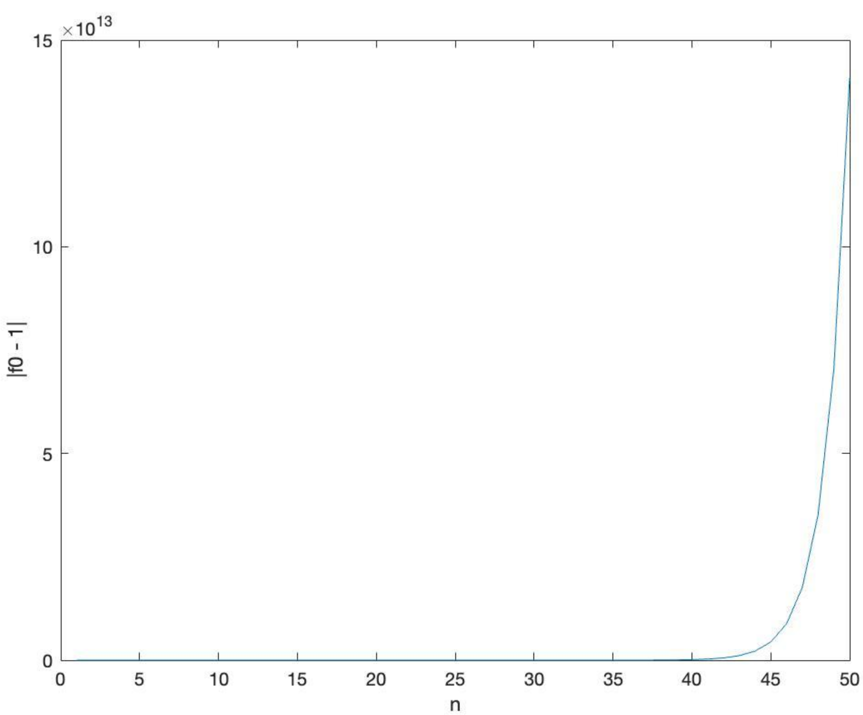
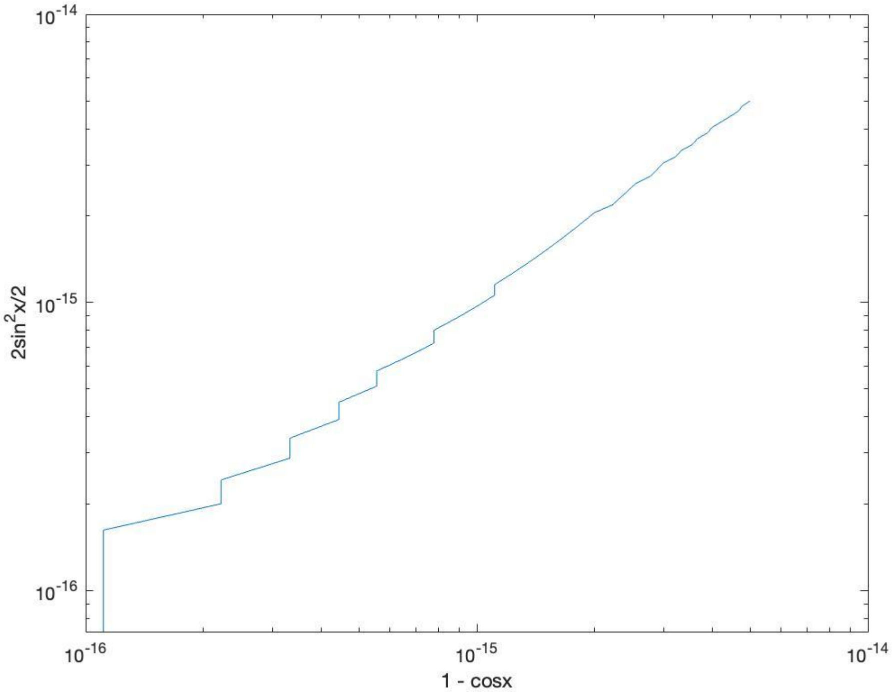
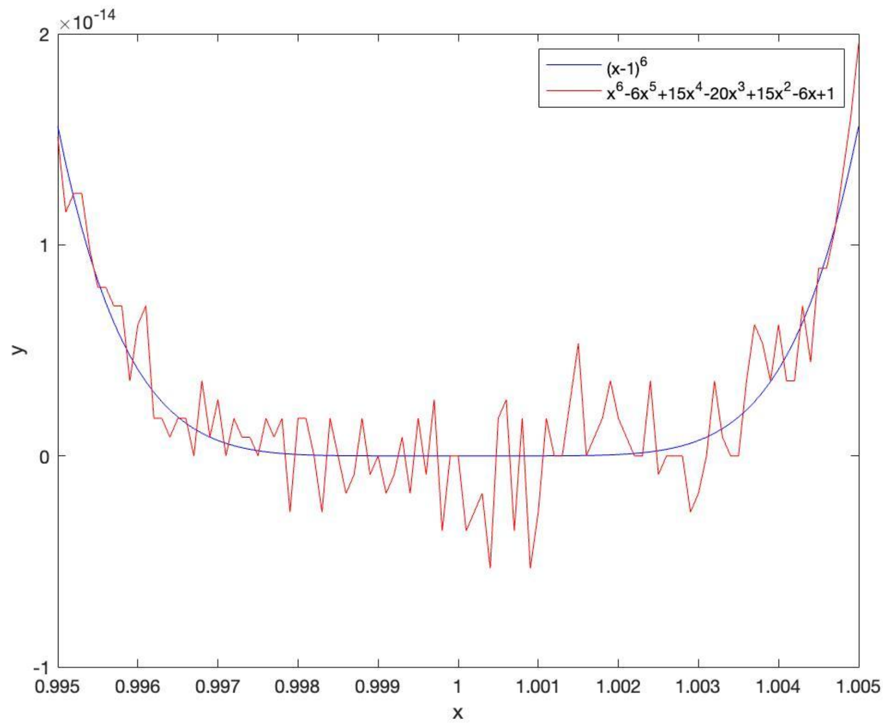


FIG:3(b)

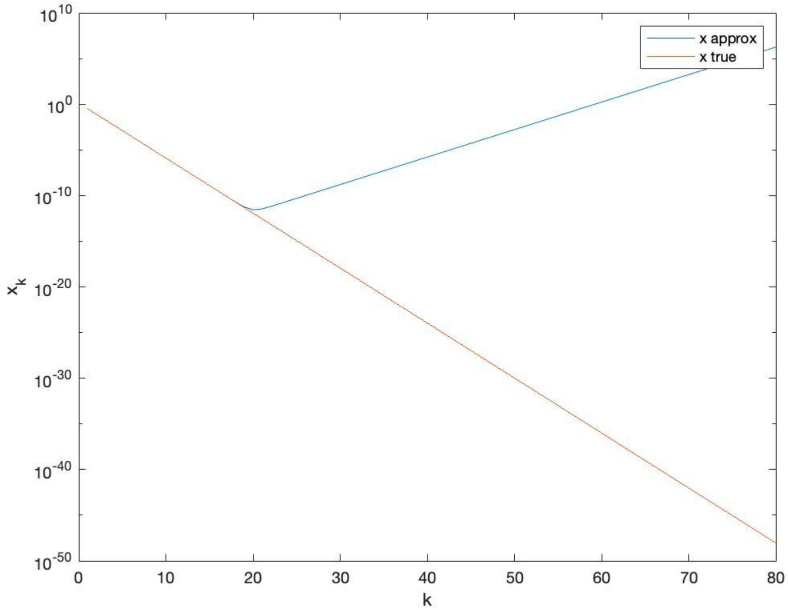
FIG :4(C)


10^{-14}
10^{-15}
10^{-16}
$2\sin^2 x/2$
10^{-16}
10^{-15}
10^{-14}
1 - cosx


$\times 10^{-14}$
2
1
0
-1
y
$(x-1)^6$
$x^6-6x^5+15x^4-20x^3+15x^2-6x+1$
0.995
0.996
0.997
0.998
0.999
1
1.001
1.002
1.003
1.004
1.005
x

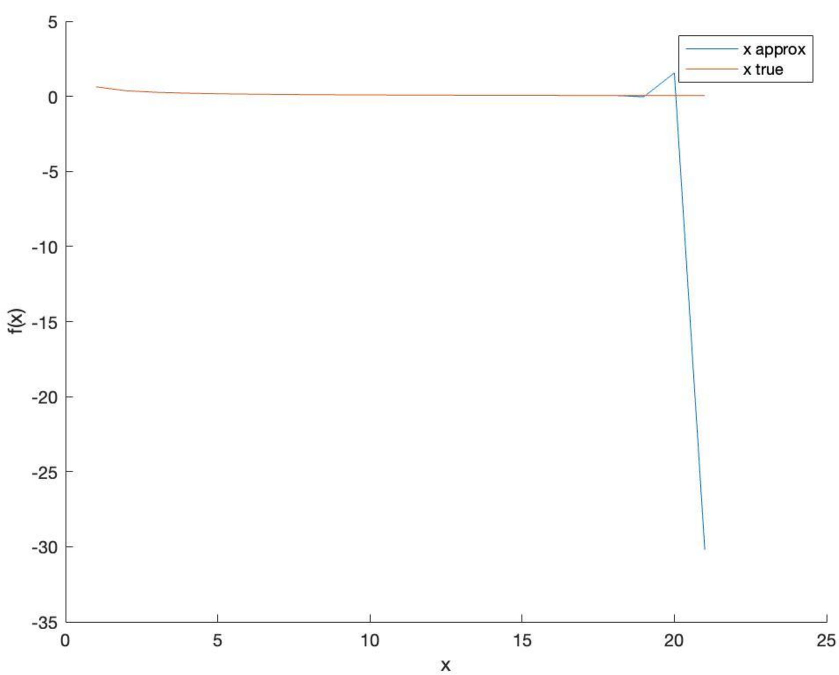
FIG(5)

FIG(6)



x approx
x true
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
10^{-40}
10^{-50}
x_k
0
10
20
30
40
50
60
70
80
k

FIG(7)



x approx
x true
5
0
-5
-10
-15
-20
-25
-30
-35
f(x)
0
5
10
15
20
25
x

```
%% q3
clc
close all
clear
max_k = 50;

% Initialize the arrays for fk and pk
fk = zeros(1, max_k);
pk = zeros(1, max_k);

% Define the value of c
c = 1 + sqrt(3) / 100;

% Define the initial values of f0 and f1
fk(1) = 1;
fk(2) = 1;

% Define the initial values of p0 and p1
pk(1) = 1;
pk(2) = 1;

% Calculate the values of fk and pk for k > 1
for k = 3:max_k
   fk(k) = fk(k-1) + fk(k-2);
   pk(k) = c*pk(k-1) + pk(k-2);
end

% Plot the values of fk and pk on a log scale plot
figure;
loglog(1:max_k, fk, '-o');
hold on;
loglog(1:max_k, pk, '-x');

% Add the lines for 1/εmach for single and double precision arithmetic
single_prec = 2^24;
double_prec = 2^53;
hold on;
loglog([1 max_k], [single_prec single_prec], 'r');
hold on;
loglog([1 max_k], [double_prec double_prec], 'g');

% Add labels and legend
xlabel('k');
ylabel('fk and pk');
```

```
legend('fk', 'pk', '1/\epsilon_{mach} for single precision', '1/\epsilon_{mach} for double precision');


max_n = 50;

% Initialize the array for f0
f0 = zeros(1, max_n);

% Define the initial values of f0
f0(1) = 1;
f0(2) = 1;

% Initialize the array for f1
f1 = zeros(1, max_n);

% Define the initial values of f1
f1(1) = 1;
f1(2) = 1;

% Calculate the values of f0 and f1 for n > 2
for n = 3:max_n
   f0(n) = f1(n-1);
   f1(n) = f0(n) + f1(n-1);
end

% Calculate the difference between the original f0 and the recomputed f0
difference = abs(f0 - 1);

% Plot the difference as a function of n
figure;
plot(1:max_n, difference);

% Add labels and legend
xlabel('n');
ylabel('|f0 - 1|');




%% q4
clc
clear
close all
x = linspace(-1e-7 , 1e-7 , 101);
```

```
y1 = 1 - cos(x);
y2 = 2*sin(x/2).^2;
loglog(y1,y2)

xlabel('1 - cosx');
ylabel('2sin^2x/2');


%% q5
clc
clear
close all
x = linspace(0.995,1.005,101);
y1 = (x - 1).^6;
y2 = x.^6 - 6*x.^5 + 15*x.^4 - 20*x.^3 + 15*x.^2 - 6*x + 1;
plot(x, y1, 'b', x, y2, 'r');
xlabel('x');
ylabel('y');
legend('(x-1)^6','x^6-6x^5+15x^4-20x^3+15x^2-6x+1')

%% q6

clc;
clear;
close all;
x1 = 1/3;
x2 = 1/12;
x = zeros(1,80);
x_true  = zeros(1,80);
x(1) = x1;
x(2) = x2;
for k = 3:80
    x(k) = 2.25*x(k-1) - 0.5*x(k-2);

end
for k = 1:80
    x_true(k) = 4^(1-k)/3;
end
semilogy(1:80, x);
hold on
semilogy(1:80, x_true);
hold off
xlabel('k');
ylabel('x_k');
```

```
legend('x approx','x true')
%% q7
clc;
clear;
close all;
i0 = 1 - 1/exp(1);
x  = zeros(1,21);
k = 1:20;
x(1) = i0;
q = zeros(1,21);
for i = 2:21
    x(i) = 1 - (i-1)*x(i-1);

end

for i = 1: 21
f = @(x) x.^(i-1) .* exp(x-1);
q(i) = integral(f,0,1);
end
hold on
plot(x)
plot(q)
xlabel('x');
ylabel('f(x)');
legend('x approx','x true')
```