Contents

Zack Humphries

COMP 521 HW1 Problem 1

```
clc;       % clear command window
clear;     % removes all saved variables
close all; % close any open windows
```

Problems a, b, & c

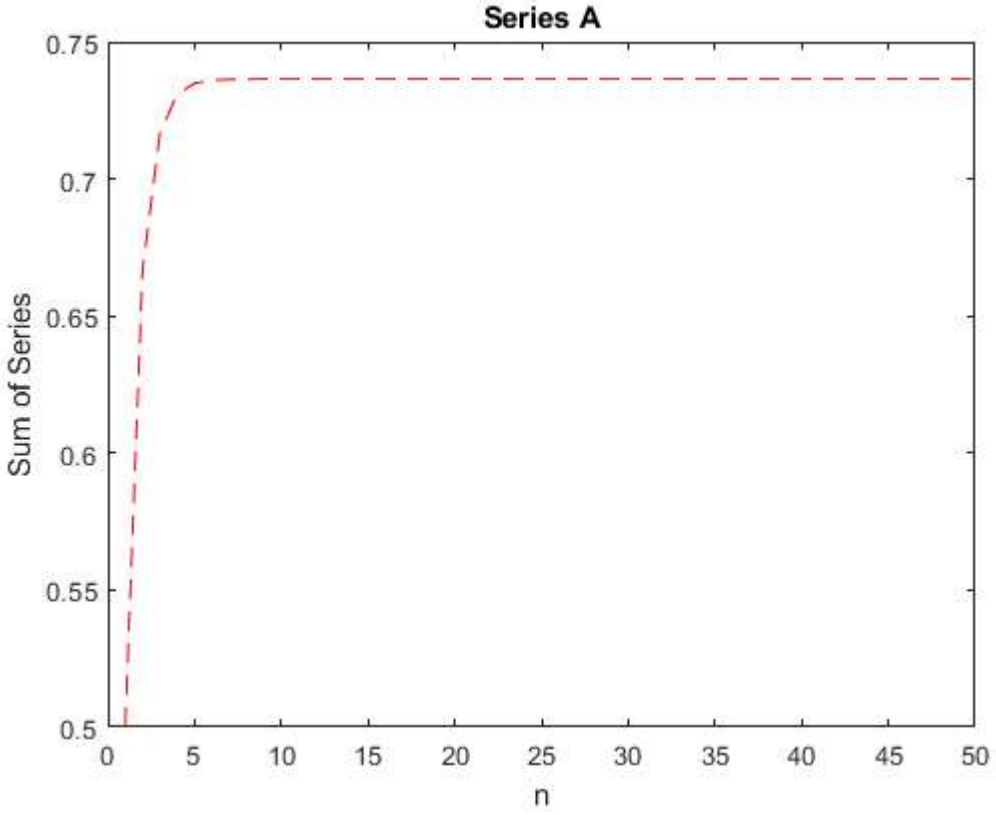
a)

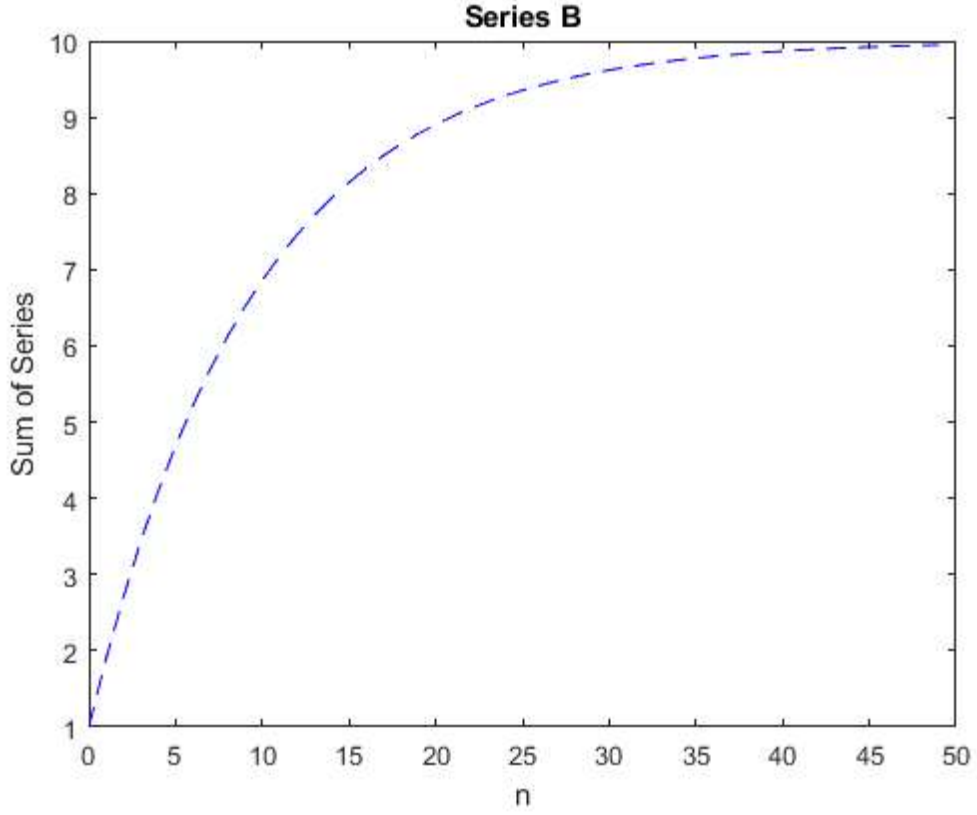
```
[finalsum_a, x_list_a, sum_list_a] = loop_problem('a', 1, 50);
plot_a = plot(x_list_a, sum_list_a, '--r');
title("Series A")
xlabel("n")
ylabel("Sum of Series")

% b)
[finalsum_b, x_list_b, sum_list_b] = loop_problem('b', 1, 50);
x_list_b = x_list_b - 1;
figure(2)
plot_b = plot(x_list_b, sum_list_b, '--b');
title("Series B")
xlabel("n")
ylabel("Sum of Series")


% c)
[finalsum_c, x_list_c, sum_list_c] = loop_problem('c', 1, 50);
figure(3)
plot_c = plot(x_list_c, sum_list_c, '--g');
title("Series C")
xlabel("n")
ylabel("Sum of Series")
```

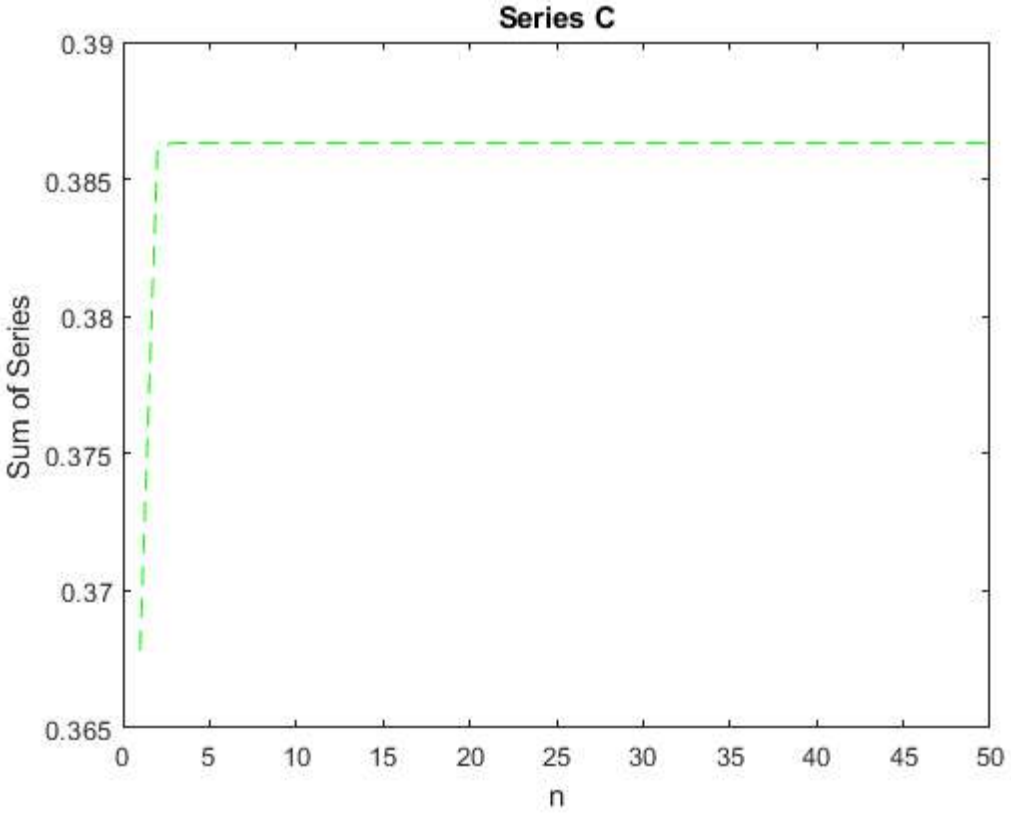
Functions Used for a, b, & c

```
function [finalsum, x_list, sum_list] = loop_problem(problem, n_start, n_end)
    x_list = [n_start: n_end];
    sum = 0.0;
    sum_list = [];

    for n=n_start: n_end
        sum = sum + function_to_use(problem, n);
        sum_list(n) = sum;
    end
    finalsum = sum;
end
```

```
function answer = function_to_use(problem, n)
    if strcmp('a', problem)
        answer = problem_a(n);
    elseif strcmp('b', problem)
        answer = problem_b(n);
    elseif strcmp('c', problem)
        answer = problem_c(n);
    end
end

function ans_a = problem_a(n)
    ans_a = ((factorial(n))^2)/(factorial(2*n));
end

%Note: Using n-1, starting at n=1 instead of using n, staring at n=0
%Is corrected for @ x_list_b = x_list_b - 1
function ans_b = problem_b(n)
    ans_b = (3^(2*(n-1)))/(10^(n-1));
end

function ans_c = problem_c(n)
    ans_c = exp(-1*(n^2));
end
```


Series A
Sum of Series
n
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25
30
35
40
45
50


Series B
Sum of Series
n
10
9
8
7
6
5
4
3
2
1
0
5
10
15
20
25
30
35
40
45
50



Published with MATLAB® R2022a

Contents

Zack Humphries

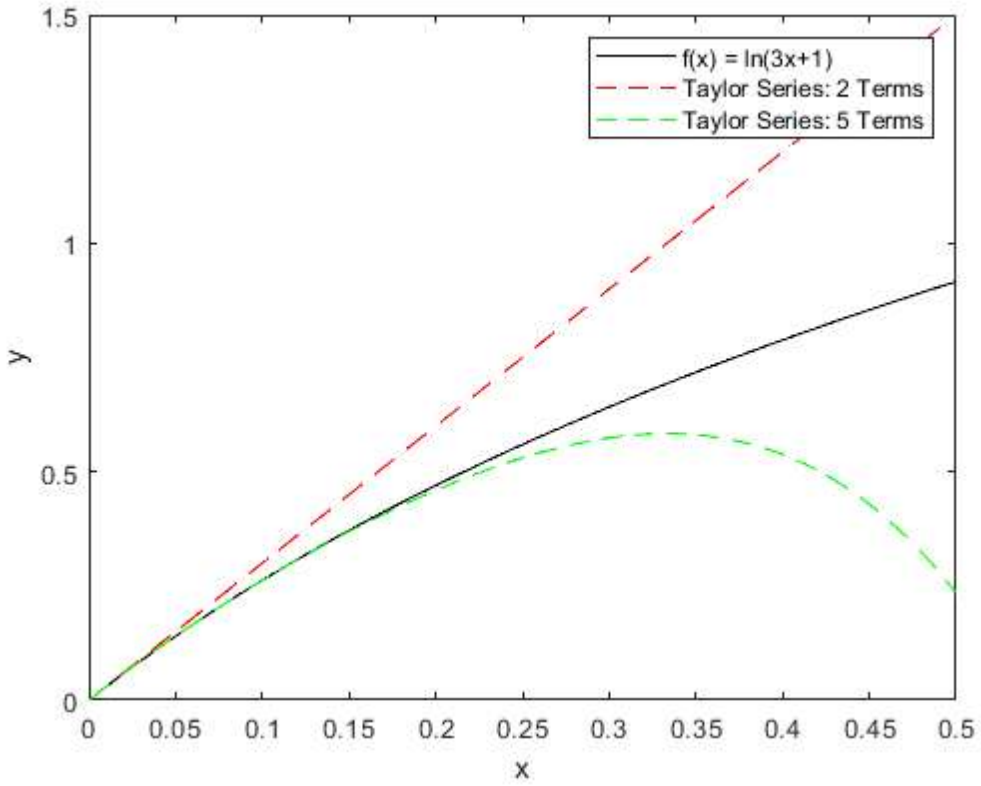
COMP 521 HW1 Problem 2

```
clc;       % clear command window
clear;     % removes all saved variables
close all; % close any open windows
```

Part 1

Both 2 and 5 terms match well with f(x), however using 5 terms is much closer to matching f(x) than just 2 terms, however, the 5 terms will trend further from f(x) the further we get away from 0=a.

```
delta_x = 0.001;
x_list = 0 : delta_x : 0.5;

% f(x) = f(a) + f'(a)(x-a)/1! + f''(a)((x-a)^2)/2! + ....

y_2_terms = y_values(x_list, 2);
y_5_terms = y_values(x_list, 5);
y_actual = actual_function(x_list);

plot(x_list, y_actual, 'k-')
hold on;
plot(x_list, y_2_terms, 'r--')
hold on;
plot(x_list, y_5_terms, 'g--')

legend("f(x) = ln(3x+1)", "Taylor Series: 2 Terms", "Taylor Series: 5 Terms")
xlabel("x")
ylabel("y")
hold off
```



Part 2

5 terms are going to have a less error than 2 terms because for each additional term, we get closer to matching the function completely

```
error_y_2_terms = y_2_terms - y_actual;
error_y_5_terms = y_5_terms - y_actual;

L2_norm_2_terms = L2_norm(error_y_2_terms)
L2_norm_5_terms = L2_norm(error_y_5_terms)

figure(2)
plot(x_list, abs(error_y_2_terms), 'r--')
hold on;
plot(x_list, abs(error_y_5_terms), 'g--')
legend("Error Vector Values: 2 Terms", "Error Vector Values: 5 Terms")
xlabel("x")
ylabel("Error Value")
hold off;
```

Functions Used for Part 1

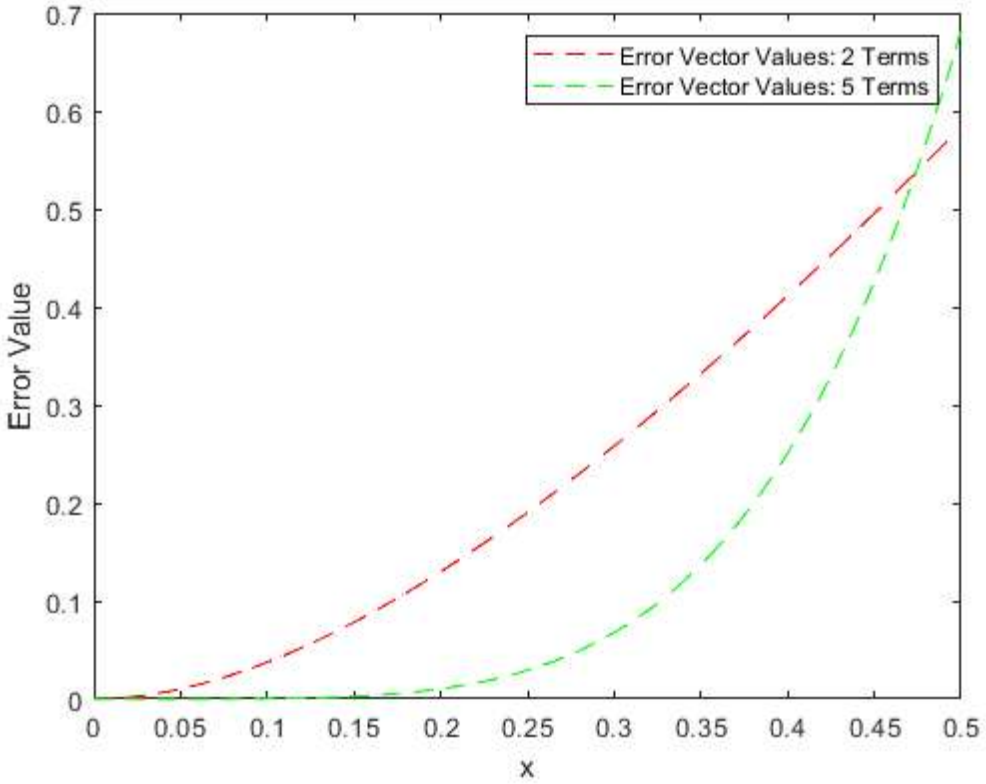
```
% Step 1: Take each x (x_list(element) and number of terms needed (terms)
% and feed into the function, get_value(...). Will eventally return with all y values
% cooresponding to that xi+h
function value_list = y_values(x_list, terms)
    value_list=zeros(1,length(x_list));
    for element=1:length(x_list)
        value_list(element) = get_value(x_list(element), terms);
    end
end
```

```
% Step 2: Feed x into the function,
% taylor_series(...). Will calculate the first 6 terms of the taylor
% series in an array. Then cuts off the array depending on how many terms
% are needed. Then sums those the terms together to get approximate f(x)
function taylor_value = get_value(x, terms)
    taylor_list = taylor_series(x);
    taylor_terms = taylor_list(1:terms);
    taylor_value = sum(taylor_terms);
end

% Step 3: Calculates taylor series for each derivative corresponding to
% f^(n)(0) and feeds to taylor_poly for rest of calculation
function taylor_list = taylor_series(x)
    f0 = taylor_poly(log(3*0 + 1),0, x);
    f1 = taylor_poly(3*((3*0 +1)^(-1)),1, x);
    f2 = taylor_poly(-9*((3*0 +1)^(-2)),2, x);
    f3 = taylor_poly(54*((3*0 +1)^(-3)),3, x);
    f4 = taylor_poly(-486*((3*0 +1)^(-4)),4, x);
    f5 = taylor_poly(5832*((3*0 +1)^(-5)),5, x);

    taylor_list = [f0, f1, f2, f3, f4, f5];
end

% Step 4: Same thing as step 3 but now also includes (x^n)/n!  (num=n)
function taylor_part = taylor_poly(equation, num, x)
    taylor_part = (1/factorial(num))*(equation*(x^(num)));
end

% Step 5: Compares to ln((3*x)+1)
function real_y = actual_function(x_list)
    real_y = zeros(1,length(x_list));
    for n=1:length(x_list)
        real_y(n) = log(3*(x_list(n)) + 1);
    end
end
```

Functions used for Part 2

```
function L2_error = L2_norm(y_terms)
    L2_error = dot(y_terms, y_terms)^(1/2);
end
```

```
L2_norm_2_terms =

    6.3944


L2_norm_5_terms =

    4.8563
```



Published with MATLAB® R2022a

Contents

Zack Humphries

COMP 521 HW1 Problem 3

```
clc;       % clear command window
clear;     % removes all saved variables
close all; % close any open windows
```

Case 1:

```
case_1_a = [1,2,-1;6,-5,4;-9,8,-7];
case_1_b = [2*pi; 5*pi; (-8)*pi];

[case_1_a_return, indx_1,d_1] = algorithm_1(case_1_a, 3);

[case_1_a_final,case_1_indx_final,case_1_X_final] = algorithm_2(case_1_a_return,case_1_b, 3, indx_1);
```

Case 2:

```
case_2_a = [pi,3*pi,2*pi;0,1,(-2/3);(-1*pi),(-3*pi),2*pi];
case_2_b = [3; 0; -1];

[case_2_a_return, indx_2,d_2] = algorithm_1(case_2_a, 3);

[case_2_a_final,case_2_indx_final,case_2_X_final] = algorithm_2(case_2_a_return,case_2_b, 3, indx_2);
```

Results

```
case_1_X_final

case_2_X_final
```

```
case_1_X_final =

    3.1416
    3.1416
    3.1416


case_2_X_final =

    0.3183
```

```
0.1061
0.1592
```

Algorithm 1

```
function [a,indx,d] = algorithm_1(a, n)

    vv = 1:n;
    d=1.0;
    iimax = 0.0;
    indx = [];

    for ii=1:n
        big=0.0;
        for j=1:n
            temp=abs(a(ii,j));
            if (temp > big)
                big = temp;
            end
        end
        if big == 0.0
            disp("rip");
            break
        end
        vv(ii)= 1.0/big;
    end

    for j=1:n
        for ii=1: (j-1)
            sum = a(ii,j);
            for k=1: (ii-1)
                sum = sum - a(ii,k)*a(k,j);
            end
            a(ii,j) = sum;
        end
        big=0.0;

        for ii=j:n
            sum=a(ii,j);
            for k=1: (j-1)
                sum = sum - (a(ii,k)*a(k,j));
            end
            a(ii,j) = sum;
            dum = vv(ii)*abs(sum);
            if dum >= big
                big = dum;
                iimax = ii;
            end
        end

        if j ~= iimax
            for k=1: n
                dum = a(iimax,k);
                a(iimax,k) = a(j,k);
                a(j,k)= dum;
            end
            d = -1*d;
            vv(iimax) = vv(j);
        end
```

```
        indx(j) = iimax;
        if a(j,j) == 0.0
            a(j,j) = 0.00001;
        end

        if j ~= n
            dum = 1.0/(a(j,j));
            for ii=(j+1):n
                a(ii,j) = a(ii,j)*dum;
            end
        end
    end
end
```

Algorithm 2

```
function [a,indx,b] = algorithm_2(a,b,n, indx)
    iii=0;
    sum = 0.0;

    for ii=1:n
        ip=indx(ii);
        sum = b(ip);
        b(ip)=b(ii);
        if iii
            for j=iii:(ii-1)
                sum = sum - a(ii,j)*b(j);
            end
        elseif sum
            iii=ii;
        end
        b(ii)=sum;
    end
    for ii=n:-1:1
        sum=b(ii);
        for j=(ii+1):n
            sum = sum - a(ii,j)*b(j);
        end
        b(ii) = sum/(a(ii,ii));
    end
end
```

Published with MATLAB® R2022a

Zack Humphries
COMP-521 (Fall 2022)

Homework 1
Problem 1

1.

a. $\sum_{n=1}^{\infty} \frac{(n!)^2}{(2n)!}$: Convergent

Ratio Test:

$$\left| \frac{[(n+1)!]^2}{[2(n+1)]!} \cdot \frac{(2n)!}{(n!)^2} \right| = \frac{(n+1)! \cdot (n+1)!}{(2n+2)!} \cdot \frac{(2n)!}{n! \cdot n!} =$$

$$\frac{(n+1)! \cdot (n+1)!}{(2n+2)(2n+1)\,(2n)!} \cdot \frac{(2n)!}{n! \cdot n!} = \left(\frac{(n+1)!}{n!}\right)^2 \cdot \frac{1}{(2n+2)(2n+1)} = \left(\frac{(n+1)n!}{n!}\right)^2 \cdot \frac{1}{(2n+2)(2n+1)}$$

$$\frac{(n+1)^2}{(2n+2)(2n+1)} = \frac{(n+1)(n+1)}{2(n+1)(2n+1)} = \frac{n+1}{4n+2} \overset{n\to\infty}{\approx} \frac{1}{4} < 1$$

b. $\sum_{n=0}^{\infty} \frac{(3)^{2n}}{10^n}$: Convergent

$$\frac{\left((3)^2\right)^n}{10^n} = \frac{9^n}{10^n}$$

Geometric Series $\sum_{n=1}^{\infty} ar^{n-1}$ $\rightarrow$ $\sum_{n=1}^{\infty} (1)\left(\frac{9}{10}\right)^{n-1}$: $\frac{9}{10} < 1$

c. $\sum_{n=1}^{\infty} e^{-n^2}$: Convergent

Ratio Test:

$$\frac{e^{-(n+1)^2}}{e^{-n^2}} = e^{-(n+1)^2+n^2} = e^{\cancel{n^2}-2n-1+\cancel{n^2}} =$$

$$e^{-2n-1} = \frac{1}{e^{2n+1}} \overset{n\to\infty}{\sim} 0$$

Problem 2

1. $f^{(0)}(x) = \ln(3x+1)$

$f^{(1)}(x) = 3(3x+1)^{-1}$

$f^{(2)}(x) = -9(3x+1)^{-2}$

$f^{(3)}(x) = 54(3x+1)^{-3}$

$f^{(4)}(x) = -486(3x+1)^{-4}$

$f^{(5)}(x) = 5832(3x+1)^{-5}$

$$f^{(0)}(x) + f^{(1)}(x)\cdot x + \frac{f^{(2)}(x)}{2!}x^2 + \frac{f^{(3)}(x)}{3!}x^3 + \frac{f^{(4)}(x)}{4!}x^4 + \frac{f^{(5)}(x)}{5!}x$$