Name: Kai Chuen Tan
Course: CSE 276C
HW#: Homework 2
Date: 17th October 2021
Professor: Dr. Christensen

1.) Prove that the first derivative $p_2'(x)$ of the parabola interpolating $f(x)$ at $x_0 < x_1 < x_2$ is equal to the straight line which takes on the value $f[x_{i-1}, x_i]$ at the point $\frac{(x_{i-1} + x_i)}{2}$, for $i = 1, 2$.

Apply Lagrange Polynomial Interpolation:

Consider the 2^{th} degree polynomial factored:

$$p(x) = a_0 L_0(x) + a_1 L_1(x) + a_2 L_2(x)$$

$$= a_0\left[\frac{x-x_1}{x_0-x_1}\left(\frac{x-x_2}{x_0-x_2}\right)\right] + a_1\left[\frac{x-x_0}{x_1-x_0}\left(\frac{x-x_2}{x_1-x_2}\right)\right] + a_2\left[\frac{x-x_0}{x_2-x_0}\left(\frac{x-x_1}{x_2-x_1}\right)\right]$$

$$= \frac{a_0}{(x_0-x_1)(x_0-x_2)}(x-x_1)(x-x_2) + \frac{a_1}{(x_1-x_0)(x_1-x_2)}(x-x_0)(x-x_2) + \frac{a_2}{(x_2-x_0)(x_2-x_1)}(x-x_0)(x-x_1)$$

$$= \frac{a_0}{(x_0-x_1)(x_0-x_2)}(x^2-(x_1+x_2)x + x_1x_2) + \frac{a_1}{(x_1-x_0)(x_1-x_2)}(x^2-(x_0+x_2)x+x_0x_2)$$

$$+ \frac{a_2}{(x_2-x_0)(x_2-x_1)}(x^2-(x_1+x_0)x + x_1x_0)$$

$$= \left[\frac{a_0}{(x_0-x_1)(x_0-x_2)} + \frac{a_1}{(x_1-x_0)(x_1-x_2)} + \frac{a_2}{(x_2-x_0)(x_2-x_1)}\right]x^2$$

$$- \left[\frac{a_0(x_1+x_2)}{(x_0-x_1)(x_0-x_2)} + \frac{a_1(x_0+x_2)}{(x_1-x_0)(x_1-x_2)} + \frac{a_2(x_1+x_0)}{(x_2-x_0)(x_2-x_1)}\right]x$$

$$+ \left[\frac{a_0(x_1x_2)}{(x_0-x_1)(x_0-x_2)} + \frac{a_1(x_0x_2)}{(x_1-x_0)(x_1-x_2)} + \frac{a_2(x_1x_0)}{(x_2-x_0)(x_2-x_1)}\right]$$

Let $A_0 = \left[\frac{a_0}{(x_0-x_1)(x_0-x_2)} + \frac{a_1}{(x_1-x_0)(x_1-x_2)} + \frac{a_2}{(x_2-x_0)(x_2-x_1)}\right]$, $A_1 = [$

Apply Newton Polynomial Interpolation:

$p_k(x) = \sum_{j=0}^{k} a_j n_j(x)$; $n_j(x) = \prod_{i=0}^{j-1}(x-x_i)$ for $j>0$; $a_j = f[x_0, x_1, \dots, x_j]$; $n_0(x) = 1$

Where $f[x_0, x_1, \dots, x_j]$ is the divided differences notation

Name : Kai Chuen Tan
Course : CSE 276C
HW# : Homework 2
Date : 17th October 2021
Professor : Dr. Christensen

1.) Continue (Part 2)

Consider the 2^{th} degree polynomial factored with Newton Polynomial Interpolation, k=2

$$p_2(x) = \cancel{\sum_{j=0}^{2} a_j \prod_{i}^{j-1}} \quad a_0 n_0(x) + \sum_{j=1}^{2} a_j \prod_{i=0}^{j-1} (x-x_i)$$

$$= a_0(1) + a_1(x-x_0) + a_2(x-x_0)(x-x_1)$$

$$= a_0 + a_1x - a_1x_0 + a_2x^2 - (a_2(x_0+x_1))x + a_2x_0x_1$$

$$p_2(x) = (a_0 - a_1x_0 + a_2x_0x_1) + (a_1 - a_2(x_0+x_1))x + a_2x^2$$

1st Derivative of $p(x)$ is the following:

$$p_2'(x) = [a_1 - a_2(x_0+x_1)] + 2a_2x^2$$

Let $i = 1$, $x = \frac{x_0+x_1}{2}$

Sub x to $p_2'(x)$

$$p_2'\left(\frac{x_0+x_1}{2}\right) = a_1 - \cancel{a_2(x_0+x_1)} + \cancel{2a_2\left(\frac{x_0+x_1}{2}\right)}$$

$$= a_1 \quad ; a_1 = f[x_0, x_1]$$

$$\boxed{p_2'\left(\frac{x_0+x_1}{2}\right) = f[x_0, x_1]}$$

Let $i = 2$, $x = \frac{x_1+x_2}{2}$

Sub x to $p_2'(x)$

$$p_2'\left(\frac{x_1+x_2}{2}\right) = a_1 - a_2(x_0+x_1) + \cancel{2}a_2\left(\frac{x_1+x_2}{\cancel{2}}\right)$$

$$= a_1 + a_2(\cancel{x_1} + x_2 - x_0 - \cancel{x_1})$$

$$= a_1 + a_2(x_2 - x_0)$$

$$= f[x_0, x_1] + f[x_0, x_1, x_2](x_2 - x_0)$$

$$= f[x_0, x_1] + \frac{f[x_1, x_2] - f[x_0, x_1]}{\cancel{x_2 - x_0}} \cancel{(x_2 - x_0)}$$

$$\boxed{p_2'\left(\frac{x_1+x_2}{2}\right) = f[x_1, x_2]}$$

~~∴ The 1st derivative $p_2'(x)$ of the pro~~

∴ The 1st derivative $p_2'(x)$ of the parabola interpolating $f(x)$ at $x_0 < x_1 < x_2$ is equal to the straight line which takes on the value $f[x_{i-1}, x_i]$ at the point $\frac{x_{i-1}+x_i}{2}$, for $i \in \{1, 2\}$

Name : Kai Chuen Tan
Course : CSE 276C
HW # : Homework 2
Date : 10/18/2021
Professor : Dr. Christensen

Problem 2

$f(x) = \cos x$; $0 \le x \le \pi$

The errors function in polynomial interpolation, $e(x)$ is defined in the following:

$$e_n(x) = f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i) \quad \text{--- } (1)$$

where, n is the degree of the polynomial

$\xi_x \in [a, b]$, which means some ξ_x is between the minimum and maximum of a, b, and x.

$p_n(x)$ is the n^{th} degree polynomial interpolating $f(x)$ at a and b.

If we use linear interpolation between adjacent points, let $p_1(x)$ be the linear polynomial interpolating $f(x)$ at x_i and x_{i+1}, $\forall\ 0 \le i \le n$.

Then, the errors function from equation (1) above can be written as follows:

$$e_1(x) = f(x) - p_1(x) = \frac{1}{2!} f''(\xi_x)\,(x - x_0)(x - x_1)$$

Since $p_1(x)$ is just an approximation of $f(x)$, $x \in [x_0, x_1]$, the error bound can be written as follows:

$$|e_1(x)| = |f(x) - p_1(x)| \le \frac{1}{2} \max_{x_0 \le x \le x_1} |f''(\xi_x)|\,(x - x_0)(x - x_1) \quad , \quad x \in [x_0, x_1] \text{ --- } (2)$$

First, maximize the following expression:

$$\max_{x_0 \le x \le x_1} |(x - x_0)(x - x_1)| = \max_{x_0 \le x \le x_1} |x^2 - (x_0 + x_1)x + x_0 x_1|$$

; $d(x) = 0$ and find x to get $\max_{x_0 \le x \le x_1} |(x - x_0)(x - x_1)|$

$$d(x) = 2x - (x_0 + x_1) = 0$$

$$x = \frac{x_0 + x_1}{2}$$

$$(x - x_0)(x - x_1) = \left(\frac{x_0 + x_1}{2} - x_0\right)\left(\frac{x_0 + x_1}{2} - x_1\right) = -\left(\frac{x_1 - x_0}{2}\right)^2$$

Let $h = x_1 - x_0$, then $(x - x_0)(x - x_1) = -\frac{h^2}{4}$

$$\max_{x_0 \le x \le x_1} |(x - x_0)(x - x_1)| = \left|-\frac{h^2}{4}\right| = \frac{h^2}{4} \quad \text{--- } (3)$$

Name: Kai Chuen Tan

Course: CSE 276C

HW#: Homework 2

Date: 10/18/2021

Professor: Dr. Christensen

Problem 2 (Part 2)

Second, maximize the following expression:

$$\max_{x_0 \le x \le x_1} \left| f''(x) \right| \quad ; \quad f(x) = \cos(x) \; ; \; f'(x) = -\sin(x) \; ; \; f''(x) = -\cos(x)$$

$$\max_{x_0 \le x \le x_1} \left| f''(x) \right| = \max_{x_0 \le x \le x_1} \left| -\cos(x) \right| = 1 \quad — \text{(4)}$$

Sub (3) & (4) to (2)

$$\left| e_1(x) \right| \le \frac{1}{2}(1)\left(\frac{h^2}{4}\right)$$

$$\left| e_1(x) \right| \le \frac{h^2}{8} \quad ; \; h = x_1 - x_0$$

To obtain 6 decimal digit accuracy,

$$\left| e_1(x) \right| \le 5 \times 10^{-7}$$

$$\frac{h^2}{8} \le 5 \times 10^{-7}$$

$$h \le 0.002$$

To obtain the number of interval, denoted as k, needed to get 6 decimal digit accuracy,

$$k = \frac{\pi - 0}{h} = \frac{\pi - 0}{0.002}$$

$$k = 1570.80 \approx 1571$$

Hence, 1572 entries are needed to obtain 6 decimal digit accuracy for the cheap robot.

Name: Kai Chuen Tan
Course: CSE 276C
HW #: Homework 2
Date: 10/18/2021
Professor: Dr. Christensen

Problem 2 (Part 3)

If we use quadratic interpolation between adjacent points, let $p_2(x)$ be the 2^{nd} degree polynomial interpolating $f(x)$ at x, x_0, x_1, and x_2.

Then, the errors function from equation ① from Part 1 of Problem 2 can be written as follows:

$$e_2(x) = f(x) - p_2(x) = \frac{1}{3!} f''(\xi_x)(x-x_0)(x-x_1)(x-x_2)$$

Since $p_2(x)$ is just an approximation of $f(x)$, $x \in [x_0, x_2]$, the error bound can be written as follows:

$$|e_2(x)| \leq \frac{1}{6} \max_{x_0 \leq x \leq x_2} |f''(\xi_x)| (x-x_0)(x-x_1)(x-x_2), \quad x \in [x_0, x_2] \text{ —— ⑤}$$

First, maximize the following expression:

$$\max_{x_0 \leq x \leq x_2} |(x-x_0)(x-x_1)(x-x_2)| \quad ;$$

Let $h = x_1 - x_0$ and $h = x_2 - x_1$

then $x_1 = x_0 + h$ and $x_2 = x_1 + h$

Consider $x = x_1 + th$,

then $x - x_0 = (x_1 + th) - (x_1 - h)$

$x - x_0 = h(t+1)$,

$x - x_1 = (x_1 + th) - x_1$

$x - x_1 = th$,

$x - x_2 = (x_1 + th) - (h + x_1)$

$x - x_2 = h(t-1)$

Lower Bound: $x_0 = x_1 + th$

$x_0 - x_1 = th$

$t = -1$

Upper Bound: $x_2 = x_1 + th$

$t = 1$

$d(x) = 0$ and find x to get $\max_{x_0 \leq x \leq x_2} |(x-x_0)(x-x_1)(x-x_2)|$

$$(x-x_0)(x-x_1)(x-x_2) = (x^2 - (x_0+x_1)x + x_0x_1)(x-x_2)$$

$$= x^3 - (x_0+x_1)x^2 + x_0x_1x - x_2x^2 + x_2(x_0+x_1)x - x_0x_1x_2$$

$$= x^3 - (x_0+x_1+x_2)x^2 + (x_0x_1 + x_0x_2 + x_1x_2)x - x_0x_1x_2$$

$$d(x) = 3x^2 - 2(x_0+x_1+x_2)x + (x_0x_1 + x_0x_2 + x_1x_2) = 0$$

$$x = \frac{2(x_0+x_1+x_2) \pm \sqrt{4(x_0+x_1+x_2)^2 - 12(x_0x_1 + x_0x_2 + x_1x_2)}}{6}$$

$$x = \frac{2(x_0+x_1+x_2) \pm \sqrt{4(x_0^2+x_1^2+x_2^2+2x_0x_1+2x_0x_2+2x_1x_2) - 12(x_0x_1+x_0x_2+x_1x_2)}}{6}$$

$$x = \frac{2(x_0+x_1+x_2) \pm \sqrt{4(x_0^2+x_1^2+x_2^2) - 4(x_0x_1+x_0x_2+x_1x_2)}}{6}$$

Name : Kai Chuen Tan
Course: CSE 276C
HW # : Homework 2
Date : 10/18/2021
Professor: Dr. Christensen

Problem 2 (Part 4)

$$x = \frac{(x_0+x_1+x_2) \pm \sqrt{x_0(x_0-x_1)+x_1(x_1-x_2)+x_2(x_2-x_1)}}{3} \quad ; \quad h = x_1 - x_0, \; h = x_2 - x_1$$

$$= \frac{(x_0+x_1+x_2) \pm \sqrt{-x_0h - x_1h + x_2h}}{3}$$

$$= \frac{(x_0+x_1+x_2) \pm \sqrt{h(x_2-x_1-x_0)}}{3}$$

$$= \frac{(x_0+x_1+x_2) \pm \sqrt{h(h-x_0)}}{3}$$

$$(x-x_0)(x-x_1)(x-x_2) = (x_0$$

$$\max_{x_0 \le x \le x_2} \left| (x-x_0)(x-x_1)(x-x_2) \right| = \max_{-1 \le t \le 1} \left| [h(t+1)]\, ht\, [h(t-1)] \right|$$

$$= h^3 \max_{-1 \le t \le 1} \left| t^3 - t \right| \quad ; \quad d(t) = 0 \text{ find } t$$

$$d(t) = 3t^2 - 1 = 0$$

$$t = \pm\sqrt{\frac{1}{3}}$$

$$t = \pm\frac{1}{\sqrt{3}}$$

$$= h^3 \max_{-1 \le t \le 1} \left| \left(-\frac{1}{\sqrt{3}}\right)^3 + \frac{1}{\sqrt{3}} \right|$$

$$\max_{x_0 \le x \le x_2} \left| (x-x_0)(x-x_1)(x-x_2) \right| = \frac{2\sqrt{3}}{9} h^3 \quad \text{— (6)}$$

Second, maximize the following expression :

$$\max_{x_0 \le x \le x_2} \left| f'''(x) \right| \quad ; \quad f''(x) = -\cos(x) \quad ; \quad f'''(x) = \sin(x)$$

$$\max_{x_0 \le x \le x_2} \left| \sin(x) \right| = 1 \quad \text{— (7)}$$

Name: Kai Chuen Tan
Course: CSE 276C
HW #: Homework 2
Date: 10/18/2021
Professor: Dr. Christensen

Problem 2 (Part 5)

Sub 6 & 7 to 5,

$$\left| e_2(x) \right| \leq \frac{1}{\cancel{6}_3}(1)\left(\frac{\cancel{2}^1\sqrt{3}}{9}\right)h^3$$

$$\left| e_2(x) \right| \leq \frac{\sqrt{3}}{27}h^3$$

To obtain 6 decimal digit accuracy,

$$\left| e_2(x) \right| \leq 5\times10^{-7}$$

$$\frac{\sqrt{3}}{27}h^3 \leq 5\times10^{-7}$$

$$h \leq 0.019827$$

To obtain the number of interval, denoted as k, needed to get 6 decimal accuracy,

$$k = \frac{\pi-0}{h} = \frac{\pi-0}{0.019827}$$

$$k = 158.45 \approx 159$$

Hence, 160 entries are needed to obtain 6 decimal digit accuracy for the cheap robot.

Contents


```
% Name          : Kai Chuen Tan
% Title         : Homework 2
% Course        : CSE 276C: Mathematics for Robotics
% Professor     : Dr. Henrik I. Christensen
% Date          : 17th October 2021

clear all;
clc;

fprintf('Name          : Kai Chuen Tan\n')
fprintf('Title         : Homework 2\n')
fprintf('Course        : CSE 276C: Mathematics for Robotics\n')
fprintf('Professor     : Dr. Henrik I. Christensen\n')
fprintf('Date          : 17th October 2021\n\n')
fprintf('--------------------------------------------------------\n\n')
```

```
Name          : Kai Chuen Tan
Title         : Homework 2
Course        : CSE 276C: Mathematics for Robotics
Professor     : Dr. Henrik I. Christensen
Date          : 17th October 2021

--------------------------------------------------------
```

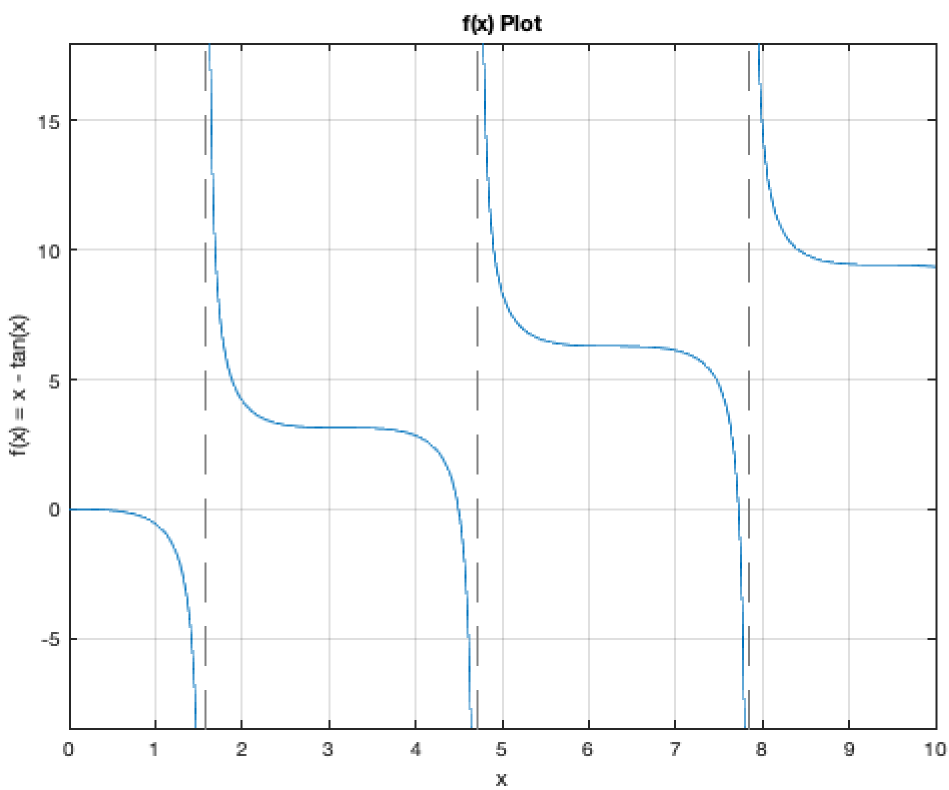
--

Problem 3 - Newton's Method

```
fprintf('Problem 3 - Newton''s Method \n')
% Given an equation x = tan (x). Find two solutions
% (upper and lower bounds) that are the nearest to x = 5.

% x = tan(x)
% 0 = x - tan(x)
% f(x) = x - tan(x)
% d(f(x))/dx = 1 - sec^2(x)

% Exact of x
x_exact = 5;
% Define x_k
x_k = (1:0.1:10);
% Define function of x
fx = @(x)x - tan(x);
% Define the 1st derivative of function x
dfx = @(x)1 - (sec(x))^2;
% Error Tolerance, e
error_tol = 1e-6;
% Maximum Iteration to quit the function
max_iter = 1000;
```

```
% Plot the graph to guess the location of the roots.
figure
fplot(fx, [0, 10]);
title('f(x) Plot')
xlabel('x')
ylabel('f(x) = x - tan(x)')
grid on

% Display two closest values.
fprintf("\nThe two solutions that are nearest to 5 are the following:\n")

% Call the Newton's Method to find two closet solutions
[x_1, x_2] = Newtons_Method(x_exact, x_k, fx, dfx, max_iter, error_tol)
```

```
Problem 3 - Newton's Method

The two solutions that are nearest to 5 are the following:

x_1 =

    4.4934


x_2 =

    7.7253
```



Published with MATLAB® R2021a

```
function [x_1, x_2] = Newtons_Method(x_exact, x_k, fx, dfx, max_iter, error_tol)

% Use Newton's Method to find the two solutions that are nearest to a solution.
% x_exact   - The exact solution
% x         - A set of x
% fx        - Function of x
% dfx       - First Derivative of fx
% max_iter  - Maximum Interation to exit the function
% error_tol - Acceptable precision of the solution

% Initialize x_vector that stores the final x_k(n) solutions
x_vector = zeros(length(x_k), 1);

for iter = 1 : length(x_k)

    % Initialize x_0
    x_0 = x_k(iter);

    % Initialize x_n
    x_n = [];

    % Calculate x_1
    x_n(1) = x_0 - fx(x_0) / dfx(x_0);

    % Start with counter 2 since x_o and x_1 were calculated
    counter = 2;

    % Apply Newton's method
    while ((abs(x_n(counter-1) - x_exact) > error_tol) && (counter <= max_iter))

        x_n(counter) = x_n(counter-1) - fx(x_n(counter-1)) / dfx(x_n(counter-1));

        counter = counter + 1;

    end

    x_vector(iter) = x_n(counter-1);

end

% Determine the difference between calculated x and exact x
difference_x = abs(x_vector - x_exact);

% Determine the minimum x
[~, index_1] = min(difference_x);

% Define array size
sorted_difference_x = unique(difference_x);

% Find the second closest solution.
for iter = 1: length(difference_x)

    if difference_x(iter) == sorted_difference_x(2)

        index_2 = iter;

        break

    end

end
```

```
% Assign to x_1
x_1 = x_vector(index_1);

% Assign to x_2
x_2 = x_vector(index_2);


end
```

Name : Kai Chuen Tan
Course : CSE 276C
HW # : Homework 2
Date : 10/17/2021
Professor : Dr. Christensen

Problem 4

Planets	Mercury	Venus	Earth	Mars	Jupiter
Distance From the Sun (10^6 km), x	58	108	149.5	227	778
Days In a Planet Year, y	88	224.7	365.3	687	4331.5

Find the function with the Lagrange Method.

$$f(x) = \sum_{i=0}^{n} y_i L_i \;;\; L_i = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

$$f(x) = \frac{(x-108)(x-149.5)(x-227)(x-778)}{(58-108)(58-149.5)(58-227)(58-778)}(88) + \frac{(x-58)(x-149.5)(x-227)(x-778)}{(108-58)(108-149.5)(108-227)(108-778)}(224.7)$$
$$+ \frac{(x-58)(x-108)(x-227)(x-778)}{(149.5-58)(149.5-108)(149.5-227)(149.5-778)}(365.3) + \frac{(x-58)(x-108)(x-149.5)(x-778)}{(227-58)(227-108)(227-149.5)(227-778)}(687)$$
$$+ \frac{(x-58)(x-108)(x-149.5)(x-227)}{(778-58)(778-108)(778-149.5)(778-227)}(4331.5)$$

$$f(x) = \frac{11(x-108)(x-149.5)(x-227)(x-778)}{69585750} - \frac{321(x-58)(x-149.5)(x-227)(x-778)}{236342500}$$
$$+ \frac{3653(x-58)(x-108)(x-227)(x-778)}{1849593009} - \frac{1374(x-58)(x-108)(x-149.5)(x-778)}{1717579955}$$
$$+ \frac{8663(x-58)(x-108)(x-149.5)(x-227)}{334113616800}$$

Name : Kai Chuen Tan
Course : CSE 276C
HW # : Homework 2
Date : 10/18/2021
Professor : Dr. Christensen

Problem 4 (Part 2)

$$f(x) = \frac{11(x^2-257.5x+16146)(x^2-1005x+176606)}{69585750} - \frac{321(x^2-207.5x+8671)(x^2-1005x+176606)}{236342500}$$

$$+ \frac{3653(x^2-166x+6264)(x^2-1005x+176606)}{1849593009} - \frac{1374(x^2-166x+6264)(x^2-927.5x+116311)}{1717579955}$$

$$+ \frac{8663(x^2-166x+6264)(x^2-376.5x+33936.5)}{334113616800}$$

$$f(x) = \frac{11(x^4-1262.5x^3+451539.5x^2-61702775x+2851480476)}{69585750}$$

$$- \frac{321(x^4-1212.5x^3+393814.5x^2-45360100x+1531350626)}{236342500}$$

$$+ \frac{3653(x^4-1171x^3+349700x^2-35611916x+1106259984)}{1849593009}$$

$$- \frac{1374(x^4-1093.5x^3+276540x^2-25117486x+728572104)}{1717579955}$$

$$+ \frac{8663(x^4-542.5x^3+102699.5x^2-7991855x+212578236)}{334113616800}$$

$$\boxed{f(x) = \frac{768646819905\,1}{87885295406786896\,20000}x^4 - \frac{3391858202199959}{7030823632542951\,69600}x^3 + \frac{1513243542455773772\,29}{1757705908135737924\,0000}x^2 + \frac{9881189111862646415\,9}{7030823632542951\,6960}x - \frac{1395061566376816496\,7}{64754859568808\,5000}}$$

Contents


```
% Name          : Kai Chuen Tan
% Title         : Homework 2
% Course        : CSE 276C: Mathematics for Robotics
% Professor     : Dr. Henrik I. Christensen
% Date          : 17th October 2021

clear all;
clc;

fprintf('Name          : Kai Chuen Tan\n')
fprintf('Title         : Homework 2\n')
fprintf('Course        : CSE 276C: Mathematics for Robotics\n')
fprintf('Professor     : Dr. Henrik I. Christensen\n')
fprintf('Date          : 21st October 2021\n\n')
fprintf('---------------------------------------------------------\n\n')
```

```
Name          : Kai Chuen Tan
Title         : Homework 2
Course        : CSE 276C: Mathematics for Robotics
Professor     : Dr. Henrik I. Christensen
Date          : 21st October 2021

---------------------------------------------------------
```

Problem 4 - Lagrange's Method

```
fprintf('Problem 4 - Lagrange''s Method\n')
fprintf('-------------------------------\n\n')

% Planets' distance from the sun, s [10^6 km]
% [Mercury, Venus, Earth, Mars, Jupiter] (left to right)
s = [58, 108, 149.5, 227, 778];
s_Uranus = 2952.4;

% Days in a Planet Year, T [days]
% [Mercury, Venus, Earth, Mars, Jupiter] (left to right)
T = [88, 224.7, 365.3, 687, 4331.5];

% Applying Vandermonde Matrix and Lagrange Polynomial to get the Lagrange
% Funcion / Equation [a_o, a_1, a_2,..., a_n]
[fx_Lagrange_coeffs] = Lagrange_Method_Eq(s,T);

% Lagrange Function Check
fx = @(x, coeffs) coeffs(1) + coeffs(2) * x + coeffs(3) * x^2 + coeffs(4) * x^3 + coeffs(5) * x^4;

% Test Lagrange function that calculate the Days in a planet year, T [days]
T_Mars = fx(s(4), fx_Lagrange_coeffs);

T_Earth = fx(s(3), fx_Lagrange_coeffs);

T_Uranus = fx(s_Uranus, fx_Lagrange_coeffs);

% Print Lagrange Function.
fprintf('The Lagrange Function is:\n\n')
fprintf('f(x) = %.4e x%c + %.4e x%c + %.4e x%c + %.4f x + %.4f\n\n',...
    fx_Lagrange_coeffs(end),8308, fx_Lagrange_coeffs(4), 179, fx_Lagrange_coeffs(3), 178, fx_Lagrange_coeffs(2), fx_Lagrange_coeffs(1))

% Print results
fprintf('Given the Mars'' distance from the Sun is %.2f million kilometers,\nthe number of days in the planet year is %.2f days.\n\n',...
    s(4), T_Mars)

fprintf('Given the Earth''s distance from the Sun is %.2f million kilometers,\nthe number of days in the planet year is %.2f days.\n\n',...
    s(3), T_Earth)

fprintf('Given the Uranus''s distance from the Sun is %.2f million kilometers,\nthe number of days in the planet year is %.2f days.\n\n',...
    s_Uranus, T_Uranus)
```

```
Problem 4 - Lagrange's Method
-------------------------------
```

```
The Lagrange Function is:

f(x) = 8.7460e-10 x⁴ + -4.8243e-06 x³ + 8.6092e-03 x² + 1.4054 x + -21.5437

Given the Mars' distance from the Sun is 227.00 million kilometers,
the number of days in the planet year is 687.00 days.

Given the Earth's distance from the Sun is 149.50 million kilometers,
the number of days in the planet year is 365.30 days.

Given the Uranus's distance from the Sun is 2952.40 million kilometers,
the number of days in the planet year is 21470.83 days.
```

Published with MATLAB® R2021a

```
function [fx_Lagrange_coeffs] = Lagrange_Method_Eq(x_vector,y_vector)
% Langrange Method Equation Function that gives the coefficients
% with given:
% x_vector - a vector with x coordinates
% y_vector - a vector with y coordinates

% Determine the length of the vector.
num_Points = length(y_vector);

% Initialize the Vandermonde Matrix, A.
A = ones(num_Points);

% Construct the Vandermonde Matrix, A.
for row = 1 : num_Points
    for column = 2 : num_Points

        A(row, column) = x_vector(row)^(column - 1);

    end
end

% Calculate the coefficients
fx_Lagrange_coeffs = A \ y_vector';

end
```

Name: Kai Chuen Tan
Course: CSE 276C
HW#: Homework 2
Date: 10/17/2021
Professor: Dr. Christensen

Problem 5

$f'(x) = e^x - 4$

$f(0) = 2$

Alice's internet speed function, $f(x)$

Use the bisection method to help Alice figure out when her internet will stop working.

$$f(x) = \int f'(x)\,dx$$
$$= \int e^x - 4\,dx$$

$f(x) = e^x - 4x + c \;;\; f(0) = 2$

$f(0) = 1 + c$

$2 = 1 + c$

$c = 1$

$f(x) = e^x - 4x + 1$

Contents


```
% Name           : Kai Chuen Tan
% Title          : Homework 2
% Course         : CSE 276C: Mathematics for Robotics
% Professor      : Dr. Henrik I. Christensen
% Date           : 17th October 2021

clear all;
clc;

fprintf('Name           : Kai Chuen Tan\n')
fprintf('Title          : Homework 2\n')
fprintf('Course         : CSE 276C: Mathematics for Robotics\n')
fprintf('Professor      : Dr. Henrik I. Christensen\n')
fprintf('Date           : 17th October 2021\n\n')
fprintf('--------------------------------------------------------\n\n')
```

```
Name           : Kai Chuen Tan
Title          : Homework 2
Course         : CSE 276C: Mathematics for Robotics
Professor      : Dr. Henrik I. Christensen
Date           : 17th October 2021

--------------------------------------------------------
```

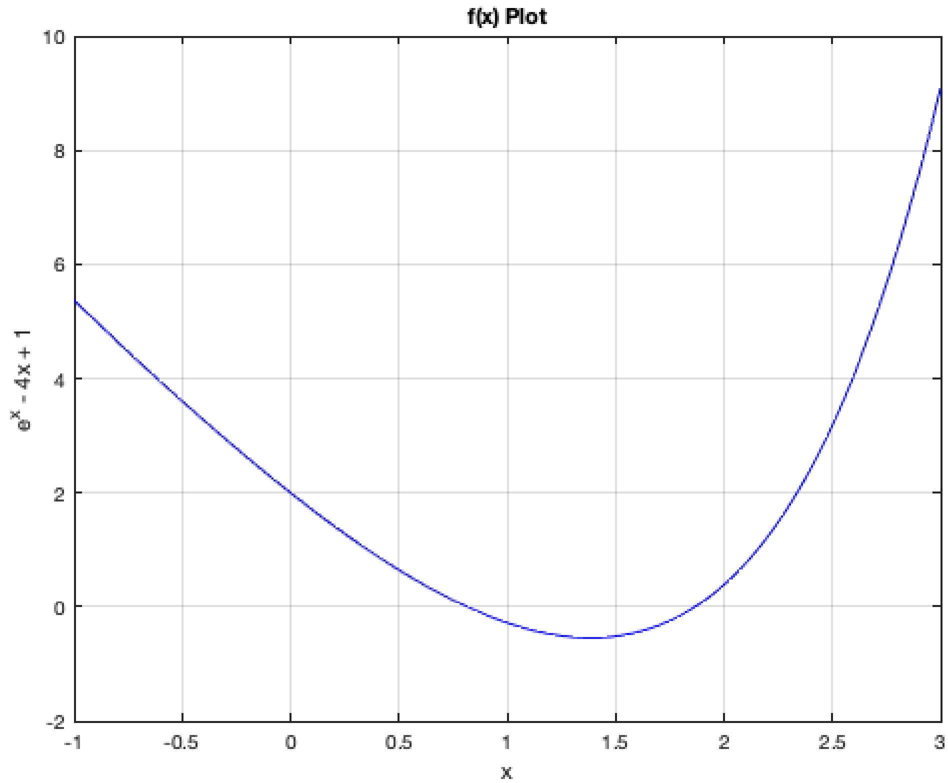
--

Problem 5 - Solving Alice's Problem with Bisection Method

```
fprintf('Problem 5 - Solving Alice''s Problem with Bisection Method \n')

% Define x
x =linspace(-1, 3, 100);
% Alice's internet speed function, f(x)
fx = @(x)exp(x) - 4 * x +1;

% Plot the graph to guess the location of the roots.
figure
plot(x,fx(x),'b-')
title('f(x) Plot')
xlabel('x')
ylabel('e^x - 4x + 1')
grid on

% Based on the plot, we know there are a root in between 0 and 1.
a_1 = 0;
b_1 = 1;

fprintf("\na is %i, and b is %i.\n", a_1, b_1)
```

```
x_root_1 = Bisection_Method( fx, a_1, b_1 )

fprintf("--------------------------------\n")

% Based on the plot, we know there are a root in between 1 and 2.

a_2 = 1;
b_2 = 2;

fprintf("\na is %i, and b is %i.\n", a_2, b_2)

x_root_2 = Bisection_Method( fx, a_2, b_2 )
```

```
Problem 5 - Solving Alice's Problem with Bisection Method

a is 0, and b is 1.

x_root_1 =

    0.8145

--------------------------------

a is 1, and b is 2.

x_root_2 =

    1.8667
```



Published with MATLAB® R2021a

```
function [ x_root ] = Bisection_Method( fx, a, b )
% Bisection Method Function
% fx is the function of x
% a and b are the constants, where x lies between them.

Fa = fx(a); % Determine f(a)
Fb = fx(b); % Determine f(b)

max_Iteration = 50; % Maximum Iteration to stop the funciton
error_tolerance = 1e-6;   % Error tolerance of the root

% If both f(a) and f(b) are both positive or both negative
if Fa*Fb > 0

    % The a and b range is invalid. Therefore, no answer.
    fprintf('a.) Error: The functions have the same sign at points a and b.\n\n')
    x_root = ('No Answer');

else

    for count = 1: max_Iteration

        % Bisection Method
        x_root = (a + b)/2;

        % Calculate error tolerance
        tolerancez = abs((b-a)/2);

        % Calculate f(x)
        FxNS = fx(x_root);

        % If f(x) = 0
        if FxNS == 0

            % x_root is the solution
            fprintf('An exact solution x =%11.6f was found.', x_root)

            % Break for loop.
            break

        % If the error tolerance is less than 1e-6
        elseif tolerancez < error_tolerance

            % Break for loop
            break

        % If the maximum counter reached
        elseif count == max_Iteration

            % Exit program
            fprintf('Solution was not obtained in %i iterations.', max_Iteration)

            % Break for loop
            break

        % If f(a)*f(b) < 0
        elseif fx(a)*FxNS < 0

            % b is the root.
            b = x_root;
        else
```

```
            % a is the root, otherwise
            a = x_root;

        end
    end
end
end
```