```
% Problem 4
close all
clc
clear
format long e
x_Bisection = bisection(10^(-10), -4.8, -4.2);
x_Newton = Newton(10^(-10),-4.2);
x_Secant = Secant(10^(-10), -4.8, -4.2);

% calculating true roots
given_fun = @(a)a - 4*sin(2*a) + 3.245892718783470;
interval = [-4.8 -4.2];
x_true = fzero(given_fun,interval);

% For Bisection Method
x_init = -4.8;
x_final = -4.2;

% computing error
for i = 1 : length(x_Bisection)

    if (x_init*x_Bisection(i)<0)
        x_final = x_Bisection(i);
    else
        x_init = x_Bisection(i);
    end

    error_Bisection(i) = abs(x_final-x_init)/2;
end

% computing Ratio
for i = 1 : length(x_Bisection)-1
    ratio_Bisection(i) = error_Bisection(i+1)/error_Bisection(i);
end
ratio_Bisection(length(error_Bisection)) = NaN;


% creating table
format long
disp('Table for Bisection Method')
T_B = table;
T_B.x_value = x_Bisection';
T_B.error = error_Bisection';
T_B.ratio = ratio_Bisection'


% For Newton Method.

% computing error
format long e
for i = 1 : length(x_Newton)
   error_Newton(i) = abs(x_true - x_Newton(i));
```

```
end

% computing ratio
for i = 1 : length(error_Newton)-1
   ratio_Newton(i) = error_Newton(i+1)/(error_Newton(i)^2);
end
ratio_Newton(length(error_Newton)) = NaN;

% creating table
format long
disp('Table for Newton Method')
T_N = table;
T_N.x_value = x_Newton';
T_N.error = error_Newton';
T_N.ratio = ratio_Newton'


% For Secant Method.

% computing error
format long e
for i = 1 : length(x_Secant)
   error_Secant(i) = abs(x_true - x_Secant(i));
end

% computing ratio
r_sec = ((1 + sqrt(5))/2);
for i = 1 : length(error_Secant)-1
   ratio_Secant(i) = error_Secant(i+1)/(error_Secant(i)^r_sec);
end
ratio_Secant(length(error_Secant)) = NaN;

% creating table
format long
disp('Table for Secant Method')
T_S= table;
T_S.x_value = x_Secant';
T_S.error = error_Secant';
T_S.ratio = ratio_Secant'
```

Table for Bisection Method

T_B =

x_value	error	ratio
-4.5	0.15	1.5
-4.65	0.225	0.833333333333333
-4.575	0.1875	0.899999999999999
-4.5375	0.16875	1.05555555555556
-4.55625	0.178125	0.973684210526315
-4.546875	0.1734375	0.986486486486486
-4.5421875	0.17109375	1.00684931506849

-4.54453125	0.172265625	1.00340136054422
-4.545703125	0.1728515625	1.00169491525424
-4.5462890625	0.17314453125	1.00084602368866
-4.54658203125	0.173291015625	1.00042265426881
-4.546728515625	0.1733642578125	1.00021123785382
-4.5468017578125	0.17340087890625	0.999894403379093
-4.54676513671875	0.173382568359375	1.00005280388636
-4.54678344726563	0.173391723632812	0.99997359945087
-4.54677429199219	0.173387145996094	0.999986799376931
-4.54676971435547	0.173384857177735	0.999993399601337
-4.54676742553711	0.173383712768555	1.00000330022111
-4.54676856994629	0.173384284973145	0.999998349894889
-4.5467679977417	0.17338399887085	0.999999174946083
-4.54676771163941	0.173383855819703	0.999999587472703
-4.54676756858826	0.173383784294129	0.999999793736265
-4.54676749706268	0.173383748531342	0.99999989686811
-4.5467674612999	0.173383730649948	1.00000005156595
-4.54676747918129	0.173383739590645	1.00000002578297
-4.54676748812199	0.173383744060994	1.00000001289148
-4.54676749259234	0.173383746296168	0.999999993554258
-4.54676749035716	0.173383745178581	0.999999996777129
-4.54676748923958	0.173383744619787	0.999999998388566
-4.54676748868078	0.173383744340391	1.00000000080572
-4.54676748896018	0.173383744480089	0.999999999597142
-4.54676748882048	0.17338374441024	NaN

Table for Newton Method

T_N =

x_value	error	ratio
-4.67810254289576	0.131335054107858	0.138223132165513
-4.54915168527989	0.00238419649199173	0.300519575778594
-4.54676919705925	1.70827134660101e-06	0.303750530653838
-4.54676748878879	8.86402062860725e-13	0
-4.5467674887879	0	NaN

Table for Secant Method

T_S =

x_value	error	ratio
-4.51354245311801	0.0332250356698962	1.78024937934442
-4.55398124891631	0.00721376012840835	0.225236875845941
-4.54669038560221	7.71031856947602e-05	0.750997426907802
-4.54676732253024	1.66257663636316e-07	0.362639814920466
-4.5467674887918	3.89466237038505e-12	0
-4.5467674887879	0	NaN

Published with MATLAB® R2016b